

Deep Learning-Based Insect Pest Detection and Classification Using Vision Transformers and Knowledge Distillation for Sustainable Agriculture

By

Anik Ahmed Rifat

213-15-4267

Md. Nirob Hossain

213-15-4245

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

Supervised by

Dr. Abdus Sattar

Associate Professor & Director, M.Sc in CSE

**Department of Computer Science and
Engineering Daffodil International University**

Co-Supervised by

Mr. Md. Sadekur Rahman

Assistant Professor

**Department of Computer Science and
Engineering Daffodil International University**



DAFFODIL INTERNATIONAL UNIVERSITY

Dhaka, Bangladesh

September 17, 2025

APPROVAL

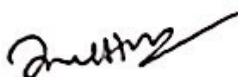
This Project titled "Deep Learning-Based Insect Pest Detection and Classification Using Vision Transformers and Knowledge Distillation for Sustainable Agriculture", submitted by Anik Ahmed Rifat, ID No: 213-15-4267 and Md. Nirob Hossain, ID No: 213-15-4245 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 17 September, 2025.

BOARD OF EXAMINERS



Dr. Sheak Rashed Haider Noori
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman



Dr. Md. Zahid Hasan
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Samia Nawshin
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Md. Arshad Ali
Professor
Department of Computer Science and Engineering
Hajee Mohammad Danesh Science & Technology
University, Dinajpur

External Examiner

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Dr. Abdus Sattar, Associate Professor & Director, M.Sc in CSE, Department of Computer Science and Engineering, Daffodil International University.** We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

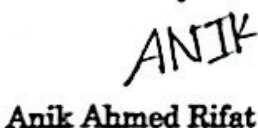


Dr. Abdus Sattar
Associate Professor & Director, M.Sc in CSE
Department of Computer Science and
Engineering Daffodil International University

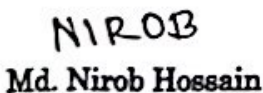
Co-Supervised by:

Mr. Md. Sadekur Rahman
Assistant Professor
Department of Computer Science and
Engineering Daffodil International University

Submitted by:



Anik Ahmed Rifat
Student ID: 213-15-4267
Department of Computer Science and
Engineering Daffodil International University



Md. Nirob Hossain
Student ID: 213-15-4245
Department of Computer Science and
Engineering Daffodil University

ACKNOWLEDGEMENTS

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

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

We are grateful and wish our profound indebtedness to **Dr. Abdus Sattar Associate Professor & Director, MSc. in CSE**, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh.

Deep knowledge and keen interest of our supervisor in the field of **Machine Learning (ML), Deep learning**, to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

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

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

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

ABSTRACT

This study proves a new pest detection system at a deep level based on deep learning technology, which promotes the productivity of farmlands through real-time pest classification. Based on state-of-the-art Vision Transformer (ViT) and Data-efficient Image Transformer (DeiT) architecture, this paper responds to the real need of an early pest prediction to avoid crop losses and also limit the use of pesticides. Applying the IP102 dataset that includes 102 different species of insect pests, special attention is paid to the six most important ones, and nearly 75,000 images are utilized in a model training. To increase the performance of the models, advanced strategies of fusion, such as the early and late fusion as well as voting within the majority, are used to enable fusion of the outputs of different models in order to obtain higher rates of accuracy in the classification. ViT/DeiT pre-trained models are fine-tuned by the usage of transfer learning methods so that limited labeled data could be used to the fullest. Accuracy, precision, recall, F1-score, and the area under the curve provide exceptionally good results whereby the Late Fusion model boasts of 98.20 accuracy and the Teacher Model (KD) with 98.33 accuracy. The Majority Voting model (97.23%) and Early Fusion model (96.68%) perform rather well as well. The study highlights the future of ensemble methods and knowledge distillation to determine the efficiency of the models and to achieve better classification results. This system will help to develop sustainable farming methods and minimize environmental impact and food security since it allows creating a scalable and resource-efficient approach to detecting pests. The future work consists in improving deployment to mobile and edge devices to have the system available to small-scale farmers in resource-constrained contexts.

Keywords: Deep Learning, Vision Transformer, Data-efficient Image Transformer, Pest Detection, IP102 Dataset, Transfer Learning, Knowledge Distillation, Ensemble Learning.

Table of Contents

APPROVAL.....	ii
DECLARATION.....	iii
ACKNOWLEDGEMENTS	iv
ABSTRACT	v
Table of Contents	vi-vii
List of Figures	viii
List of Tables	ix
Chapter 1	1
Introduction.....	1
1.1 Introduction	1
1.2 Motivation.....	2
1.3 Objectives.....	2
1.4 Research Question.....	3
1.5 Project Outcome.....	4
1.6 Organization of the Report.....	4
Chapter 2	5
Background	5
2.1 Introduction	5
2.2 Literature Review.....	5
2.2.1 Similar Applications	10
2.3 Gap Analysis.....	10
2.4 Summary	11
Chapter 3	12
Research Methodology	12
3.1 Methodology.....	12
3.1.1 Overview	12
3.1.2 Proposed Methodology	12
3.2 Detailed Methodology and Design	13
3.2.1 Ensemble Fusion Approaches Using ViT and DeiT for Improved Transfer Learning	14
3.2.2 Optimized Knowledge Distillation for Model Building: A ViT- EfficientNetB4 Hybrid Approach	17
3.2.3 Pre-processing and Dataset Creation.....	20
3.2.4 Algorithm Workflow	25
3.2.5 Model Architecture	26
3.3 Project Plan	38
3.3.1 Project Plan Phase	39
3.3.2 Research Resources	40
3.3.3 Risk Management.....	40
3.3.4 Expected Deliverables	41

3.4 Task Allocation	41
3.5 Summary	42
Chapter 4	44
Implementation and Results.....	43
4.1 Environment Setup	43
4.1.1 Hardware Requirements	43
4.1.2 Software Requirements	43
4.1.3 Frameworks and Libraries	43
4.1.4 Dataset Handling and Preprocessing Tools	44
4.1.5 Version Control and Collaboration.....	44
4.1.6 Model Training Configuration.....	44
4.1.7 Evaluation Setup	46
4.1.8 Working and Communication Tools	46
4.2 Testing and Evaluation/Performance/ Comparative Analysis	46
4.3 Results and Discussion.....	64
4.4 Summary	65
Chapter 5	66
Engineering Standards and Design Challenges.....	66
5.1 Compliance with the Standards.....	66
5.1.1 Software Standards	66
5.1.2 Hardware Standards	67
5.1.3 Communication Standards.....	67
5.2 Impact on Society, Environment and Sustainability.....	68
5.2.1 Impact on Life	69
5.2.2 Impact on Society & Environment	69
5.2.3 Ethical Aspects	70
5.2.4 Sustainability Plan	71
5.3 Project Management and Financial Analysis	71
5.4 Complex Engineering Problem	75
5.5 Summary	79
Chapter 6	80
Conclusion	80
6.1 Summary	80
6.2 Limitation	80
6.3 Future Work	81
References.....	82

List of Figures

Figure 3.1: Full Workflow	13
Figure 3.2: Proposed Methodology for Ensemble Fusion Approaches Using ViT and DeiT for Improved Transfer Learning	16
Figure 3.3: Proposed Methodology for Optimized Knowledge Distillation for A ViT-EfficientNetB4 Hybrid Approach	19
Figure 3.4: Proposed Data Preprocessing	20
Figure 3.5: All Classes Images	21
Figure 3.6: Proposed Architecture of Ensemble Fusion Approaches Using ViT and DeiT	31
Figure 3.7: Optimized Knowledge Distillation Architecture for Model Building: A ViT-EfficientNetB4 Hybrid Approach	38
Figure 4.1: Early Fusion and Late Fusion: Training Loss vs Validation Loss	48
Figure 4.2: ViT Teacher Model and EfficientNetB4 Student Model: Training vs Validation Loss and Teacher vs Student Models: Loss Comparison	49
Figure 4.3: Confusion Matrix of Early Fusion	53
Figure 4.4: Confusion Matrix of Late Fusion	55
Figure 4.5: Confusion Matrix of Majority Voting Fusion	56
Figure 4.6: Confusion Matrix of Teacher Model (KD)	57
Figure 4.7: Confusion Matrix of Student Model (KD)	58
Figure 4.8: ROC Curve of Early Fusion, Late Fusion, Majority Voting, and Chance.	59
Figure 4.9: ROC Curve of Teacher Model (KD)	60
Figure 4.10: ROC Curve of Student Model(KD)	61
Figure 4.11: Comparison of accuracy between all models.	62
Figure 4.12: Comparison of loss between all models.	63
Figure 4.13: Web Application.	65

List of Tables

Table 2.1: Summary of Literature Reviewed	8
Table 2.2: Gap Analysis	12
Table 3.1: Principal Activities in each period of the project	42
Table 4.1: Classification report of Early Fusion	50
Table 4.2: Classification report of Late Fusion	51
Table 4.3: Classification report of Majority Voting.	52
Table 4.4: Classification report of Teacher Model.	53
Table 4.5: Classification report of Teacher Model.	53
Table 5.1: Cost Analysis and Budget Estimation for Deep Learning-based Pest Detection System	72
Table 5.2: Mapping with Complex Engineering Problem.	75
Table 5.3: Mapping with knowledge Profile.	76
Table 5.4: Mapping with Complex Engineering Activities.	78

Chapter 1

Introduction

This chapter presents the research project with its aims, aims and scope, and its importance. It gives a review of the problematic situations that exist when detecting pests in agriculture, and the application of deep learning models, namely, Vision Transformer (ViT) and Data-efficient Image Transformer (DeiT), to solving them. Among the things which this chapter brings out are objectives of the research, the rationale behind the research and how the research would contribute in enhancing pest management in agriculture.

1.1 Introduction

Agriculture is the foundation of global food security but is challenged with various challenges by pest infestations that cause crop loss and reduce yields. Sustainable agricultural productivity necessitates successful pest control. Traditional pest identification methods, in which identification is manual or by an expert, are laborious, prone to human error, and actually not viable for large-scale agriculture. In recent years, pest crop loss has been growing alarmingly high globally, and it is estimated that pests are responsible for up to 40% of crop yield loss every year [1]. This necessitates quick automated, accurate, and scalable pest classification solutions. Automated systems can help reduce the farmer's burden by facilitating real-time, reliable, and effective pest identification techniques that improve pest management [2].

This study is aimed at the creation of a deep learning-based classification system of pests to help mitigate crop loss and enhance agricultural productivity. Using the IP102 dataset with more than 75,000 insect images of 102 insects [4], the research will focus on six insect pests, namely, *Pieris canidia*, *Potosiabre vitarsis*, grub, small brown plant hopper, yellow cutworm and yellow rice borer, which are the key crop pests [5]. The system discusses the Vision Transformer (ViT), Data-efficient Image Transformer (DeiT) and ensemble models to identify precise pests. Early fusion takes the output of both ViT and DeiT models to improve the performance [6], and late fusion applies the two-output following the classification [7]. Also, ensemble learning method can increase the robustness and accuracy of the system by a voting system [8].

In this study, a new concept of knowledge distillation is presented, where ViT is the teacher model and EfficientNetB4 is the student model. Knowledge distillation lets the student model to learn through the teacher, improving performance without adding computational costs. This method is useful in transferring knowledge, which is often

found in large models to smaller and more efficient models, thus they can be applied to detect pests in real-time in agriculture [9]. The models will be tested using data on IP102 that concentrates on 5, 10, and 15 insect pest classes. With the imbalance in the dataset, it is a good challenge to test the accuracy of the models [10]. The aim is to obtain high classification accuracy with the models being computationally viable in the large-scale agricultural applications. The paper will seek to present an automated and scalable system of insect identification to enhance pest control, minimize crop losses and increase food security [11].

1.2 Motivation

Agriculture plays an important role in food security across the world, and the attacks of pests result in massive losses as crops and yields are damaged, putting food production and the livelihoods of farmers at stake. Sustainable agriculture is threatened by pests which continue to contribute to up to 40 percent of world crop losses despite the advances in agricultural practices. Conventional methods of pest detection, which use manual observation or professional expertise, are labor intensive, prone to error and in large-scale farming, they are increasingly difficult to detect, as pests become increasingly diverse. Deep learning methods such as Convolutional Neural Networks (CNNs) have been created in response to this and can be used to perform automated classification of pests, thereby alleviating the workload on farmers and professionals. The hidden patterns in large pest images can be detected through these models, which results into fast and sound detection. The proposed study will design an effective, automatic pest-classification model that utilizes the Vision Transformers (ViT), Data-efficient Image transformer (DeiT), and ensemble classifiers (voting and knowledge distillation). Knowledge distillation enables the training of a small EfficientNet B4 model with a large ViT model, making the system more computationally and resources-constrained-friendly to deploy to resource-limited rural locations. The long-term objective will involve ensuring farmers will be able to identify pests at the initial stage, decrease loss of crops, cut down pesticide application, and foster sustainability in agriculture. The study will enhance pest management and AI-based food security in the face of rising global food need and climate alteration.

1.3 Research Objectives

Based on our research, our main objectives are mentioned as follows.

- To Develop a deep learning algorithm for detecting and classifying insect pests.
- To Employ improved image processing technology in pest detection.
- To Detect and identify a large number of pest species with high accuracy.
- To Compare the accuracy of the proposed approaches with existing solutions.
- To Emphasize the applicability of the solution to real-world conditions.

- To Test the scalability and efficiency of the approaches for large-scale farming.
- To Provide an environmentally friendly solution to pest problems.
- To Reduce the usage of harmful pesticides in farming.
- To Ensure the system is scalable, adaptable, and affordable.
- To Make the system easy to install and implement in various agricultural environments with minimal resources.
- To Acknowledge that further research is needed to improve the system.
- To Aim to enhance system precision for different pest types and land conditions.

1.4 Research Questions

Here are two research questions derived from based on our research:

RQ1: How effectively can a deep learning-based algorithm detect and classify diverse insect pest species with high accuracy under real-world agricultural conditions?

RQ2: How does the proposed approach compare with existing solutions in terms of accuracy, environmental impact, and reduction of harmful pesticide usage?

1.5 Project Outcome

- Development of an advanced deep learning-based framework for reliably recognizing and categorizing insect pests.
- Utilization of state-of-the-art models, including Vision Transformers (ViT) and Data-efficient Image Transformers (DeiT), with early and late fusion methods and model ensembles based on a voting mechanism to achieve high classification accuracy for six pest species in the IP102 dataset.
- Application of knowledge distillation to enhance performance, transferring knowledge from a larger ViT teacher model to a more efficient EfficientNetB4 student model, ensuring faster and more resource-efficient classification without compromising accuracy.
- Testing the system's efficacy for real-time pest detection, offering a scalable solution for early pest identification, which can help farmers detect pests early.
- Potential benefits include increased agricultural output, long-lasting pest control, and reduced crop wastage, ultimately improving food security.
- The system will be evaluated for adaptability, cost-effectiveness, and environmental sustainability, presenting a viable solution for large-scale agricultural pest detection.

1.6 Organization of the Report

Chapter 1: Introduction

In this chapter, the author states the background, purposes, scope and importance of the research. It also talks about the motivation of the study as well as the relevance of deep learning models, especially Vision Transformer (ViT) and Data-efficient Image Transformer (DeiT) within the pest-detecting model of agriculture.

Chapter 2: Background

The chapter is a literature review of the available concepts on the research topic, with an extensive discussion on the detection of pests in agriculture, application of deep learning methods and how ViT and DeiT models apply to relevant procedures.

Chapter 3: Methodology of Research

In this chapter, the research design and the methodology used were explained along with data collection processes. It further discusses the process of incorporating models of ViT and DeiT, how these models have been preprocessed and, how the fusion procedures (early fusion, late fusion and majority voting) have been implemented to increase the accuracy of classifications.

Chapter 4: Result and Implementation

The details of the system setup and configuration are presented in this chapter such as the hardware, software, and software tooling employed in both training and evaluation of the models. The model performance results are given, and models results are compared against each other, including accuracy, precision, recall, f1-score and AUC.

Chapter 5: Standard and design complexity in engineering.

In this chapter, the engineering standards of the project will be discussed e.g. software and hardware and communication standards. It also points out the design issues that have been involved in making the models practical to use in real time pest detection and the need to balance the accuracy of the models with the hardware resource they would use in the models and making the model deployable in resource limited spaces.

Chapter 6: Conclusion

The last chapter concludes the results of the study, comments on the work done and explains what additional research can be done to enhance and increase the study. It identifies the limitations as well and recommends how some challenges can be overcome in future versions of the pest detection system.

Chapter 2

Background

The chapter gives a clear discussion about the background knowledge of the research namely, the reading and reading of all the available methods of detection of pests, depth learning, and usage of the Vision Transformer (ViT) and Data-efficient Image Transformer (DeiT) models in doing the classification of the images among others. It also addresses the issues of insect pest classification, the role of pest detection in agriculture and the value of applying advanced deep learning models towards such an end.

2.1 Introduction

Detecting insect pests plays a key role in the safeguarding of agricultural produce and food security. The conventional approaches are based on manual processing, which involves professional competencies and is time-intensive, which is unrealistic in terms of large-scale application. As the influence of pests increases, the use of deep learning (DL) and computer vision methods of automated detection has become popular since it is accurate, scalable, and efficient. Pest detection has been transformed by Convolutional Neural Networks (CNNs) which can extract features of raw images and enhance the accuracy of the classification. The intricacy of pest identification has however, been caused by cases of similar species and environmental deviations and this has resulted in the development of novel models such as dense net and vision transformers (ViTs). Ensemble techniques and knowledge distillation have also proven to be effective in improving model performance. The literature review presents the most prominent developments, difficulties, and new tendencies in pest detection with the help of DL. It talks about the significance of model architectures like CNNs and ViTs and the significance of datasets like IP102 to train these models. The study also discusses the techniques applied in pest detection, the achievements, and gaps that can be filled by researchers in the future.

2.2 Literature Review

On-farm pest detection plays an important role in the assurance of crop output and food security. Conventional techniques are too labor intensive, imprecise and cannot be used at a large scale. Since the advent of deep learning (DL) and computer vision, automated pest detection is more precise, scalable and efficient. Convolutional Neural Networks (CNNs) have transformed the field of pest detection by deriving hierarchical features on

raw pictures. Initial studies have used CNN architecture VGG19 and ResNet50 to identify pests, with X. Wu et al. [12] reporting 51.30 and 52.73 percent higher image processing performance by CNNs compared to conventional image processing. In a parallel manner, W. Liu et al. [13] revealed that ResNet50 is highly accurate in determining the classification of pest species. Such CNN models have become common in automated pest classification, and are likely to keep improving. But as the challenge of detecting pests increased due to the presence of similar-looking species and variations in environmental conditions, the flaw of traditional CNN architectures became apparent. As a remedy to such problems, researchers began exploring higher-order architectures like DenseNet121 [14], which improved inter-layer information flow and facilitated better gradient flow during learning.

DenseNet121's ability to strongly connect all the network layers led to improved accuracy in classifying pest species with complex morphological variation. More recently, Vision Transformers (ViTs) have emerged as a robust alternative to CNNs, particularly due to their ability to learn long-range dependencies in images. ViTs are in a better position to learn spatial relationships between images and are thus extremely well-suited for use in applications such as pest detection. Kasinathan et al. [15] explored the application of ViTs in insect pest detection and reported promising results in detecting various species of pests. ViTs outperform traditional CNNs in cases of large-scale image data with complex features. Moreover, DeiT has been integrated with ViTs to reduce computational overhead without compromising accuracy. ViT integrated with DeiT is becoming the trend in pest detection systems where inference times must be pushed down and resource utilization reduced [16]. Ensemble methods, where many models have been employed in making the predictions more robust, have also been employed in pest detection. Researchers have employed CNNs and ViTs jointly to leverage both the models' strengths.

Nanni et al. [17] employed a combined method that utilized different CNN models so as to yield improved performance in pest classification. Ensemble learning is particularly beneficial in pest detection as it allows models to break the particular constraints of individual methods, leading to a more accurate and trustworthy system. An approach of this nature has been shown to be better than single-model designs, particularly when applied to datasets with a diversity of insect species. Evidence from Nanni et al. [18] and other research has confirmed the power of knowledge distillation in achieving more lightweight models without loss in performance. One of the key aspects of deep learning algorithms for pest detection is the need for model effectiveness, especially when it comes to real-time use in agriculture. Distillation, or how a smaller "student" model is taught by a larger "teacher" model, is one of the strategies that have gained popularity because of this. In pest detection, very large models like ViTs can even be employed as teacher models to train small models like EfficientNet B4 to learn the same representations but

with reduced computation intensity. The approach has proved effective in maintaining the high accuracy while reducing the training and inference time and resources. The dataset has played a major role in the creation of deep learning applications to pest detection through IP102, comprising more than 75000 images of 102 pest species. The magnitude and variety have made it possible to train generalizable and accurate models. Nevertheless, intra-class variability in which pests of the same species may vary with age, sex, and environment has been a challenge. This has brought more advanced models like the hybrid and ensemble models [12]. In the recent studies, hybrid fusion models, which are composed of several deep learning techniques such as CNNs and ViTs, have been studied to enhance the accuracy of pest detection, particularly when different pest features are presented. Merging model features or combining results post-training has been done using fusion techniques such as early and late fusion to increase detection performance [19]. One of the challenges is that it is necessary to have efficient real-time models capable of managing large bodies of data.

EfficientNet is a lightweight design, which has been commonly used to tradeoff between high accuracy and inexpensive computing. EfficientNet B4 has demonstrated to be especially an effective student model in knowledge distillation frameworks, and has been shown to enhance performance and efficiency [20]. Significant progress in insect pest detection has been enabled by the deep learning models, specifically CNNs, ViTs, and fusion with other techniques. Knowledge distillation, ensemble learning and model fusion have yielded greater accuracy and efficiency and are involved in real world applications in agriculture. These models have been trained and validated using the IP102 dataset. Although this has improved, issues such as variability of classes and computational efficiency do still exist and future studies are probably to be aimed at optimization of models in practice. Table 2.1 is a summary of the major studies, methodologies and results in the area of pest classification and knowledge distillation in agriculture.

Table 2.1: Summary of Literature Reviewed.

Author(s)	Year	Title	Methodology	Key Findings
Y. Liu et al.[21]	2016	A Study on Insect Pest Recognition Using CNNs	Deep Learning (CNN)	Achieved high classification accuracy using CNN models like VGG19 and ResNet50 for pest detection.
W. Liu et al.[13]	2016	Use of ResNet50 for Agricultural Pest Classification	Deep Learning (CNN)	Demonstrated the effectiveness of ResNet50 for classifying agricultural pests with high precision.
Y. Liu et al.[23]	2016	SSD: Single Shot MultiBox Detector	Deep Learning (SSD MobileNet)	Used SSD for real-time pest classification with an accuracy of 90.6% for five pest classes.
Nanni et al.[18]	2020	Bio-Inspired Models for Insect Pest Classification	Deep Learning (CNN Ensemble)	Showed significant improvements in accuracy by combining multiple CNN models for pest classification.
Wang et al.[21]	2020	MS-CapsNet for Pest Recognition	Deep Learning (CapsNet)	Employed multi-scale capsule networks for pest identification with a focus on size-variant species.
Nanni et al.[17]	2020	Bio-Inspired Insect Pest Classification	Deep Learning (Bio-Inspired)	Achieved 92.4% accuracy in classifying insect pests using bio-inspired models for feature extraction.
Noor et al.[25]	2020	Transfer Learning for Insect Pest Recognition	Deep Learning (Transfer Learning)	Utilized transfer learning for pest classification, reaching 89.6% accuracy for eight insect species.

Author(s)	Year	Title	Methodology	Key Findings
Kasinathan et al.[15]	2021	Hybrid Approach for Pest Detection Using ViT	Deep Learning (ViT and CNN)	Combined ViT and CNN for pest detection, achieving improved accuracy in large-scale pest classification.
Chen et al.[9]	2021	Detecting ripe fruits under natural occlusion and illumination conditions	Deep Learning (CNN)	Focused on real-time insect pest detection, achieving 80.3% precision with the AlexNet model.
Mohamed et al.[24]	2021	Mobile Application for Pest Recognition using Deep Learning	Mobile App and Deep Learning	Developed a mobile app with Faster R-CNN, achieving 98% accuracy for five pest classes.
Xu et al. [22]	2022	Multi-scale Convolution-Capsule Network for Pest ID	Deep Learning (CNN and Capsule)	Proposed a multi-scale capsule network, achieving 89.6% accuracy for pest identification.

2.2.1 Similar Applications

One of the major methodologies in this study would be case studies and mobile applications which use deep learning and image processing to identify and classify pests. The Faster R-CNN-based mobile app was created by Mohamed et al. (2021), who demonstrated 98% accuracy in recognising five pest species in real-time on mobile devices, which proves the viability of pest classification through the use of deep learning models on mobile devices [10]. In this paper, Li et al. (2018) developed a CNN-based algorithm that can identify a pest in a rice field with 82.4 percent accuracy to guide the model architecture decisions [26]. Y. Liu et al. (2019) applied a real-time pest detector using the Single Shot Multibox Detector (SSD) and MobileNet with an accuracy rate of 90.6% that is necessary in low-resource mobile systems in the agricultural area [9]. Nanni et al. (2020) combined multiple CNN-based models with bio-inspired methods and achieved 92.4% accuracy, highlighting the importance of the ensemble learning approach to enhancing the accuracy in classification [27]. Li et al. (2020) applied YOLOv5-S to real-time pest detection with a 89.6 percent accuracy, and discussed how to optimize detection speed [28]. These investigations will inform the creation of more effective scalable systems of detecting pests.

2.3 Gap Analysis

Existing models of pest detection, including CNNs, Faster R-CNN, and YOLO, are typically trained using small datasets or just one to several popular pest species, which limits their application in practice in agricultural contexts that require simultaneous detection of several species. Moreover, the performance of many models is not generally applicable, and the performance declines due to the application to large and more diverse datasets such as IP102. The purpose of this research is to fill in these gaps by formulating a more powerful model that can categorize a large range of pests in the IP102 dataset (102 species) through advanced fusion methods, like early and late fusion and knowledge distillation. The prevailing literature ignores basic computation efficiency and scaling aspects of pest detection models, focusing on accuracy but not processing speed, as is important in real-time and large scale systems. This paper fills this gap with the optimisation of models to execute effectively on resource constrained devices such as mobile phones and embedded systems. Although there has been an interest in ensemble techniques such as voting, not many studies have tested models such as ViT, DeiT, and EfficientNet in conjunction to classify pests. The present paper suggests new ensemble strategies and employs knowledge distillation (with ViT as a teacher and EfficientNet-B4 as the student) to enhance performance without introducing extra computational complexity. Overall, the purpose of this research is to address the deficiencies in pest detection to improve accuracy, generalizability, computational performance, and make use of new ensemble methods on a large-scale agricultural system.

Table 2.2: Gap Analysis

Features	TechLand bd	Ryan s	Computer Village	StarTec h	Paragon- Computer bd	Proposed system
Like or dislike to products	No	No	No	No	No	Yes
Filtering liked and disliked products	No	No	No	No	No	Yes
Add to favorite or wishlist	Yes	Yes	Yes	Yes	Yes	Yes
Search option of products	Yes	Yes	Yes	Yes	Yes	Yes
Detailed descriptions of products	Yes	Yes	Yes	Yes	Yes	Yes
Offers collection	Yes	Yes	Yes	Yes	Yes	Yes
Customer reviews and ratings	Yes	Yes	Yes	Yes	Yes	Yes
Multiple payment options	Yes	Yes	Yes	Yes	Yes	Yes
FAQs option	No	Yes	Yes	Yes	No	Yes
Chatting option	Yes	Yes	Yes	No	Yes	No
Recommendations or filtering latest products	Yes	Yes	Yes	Yes	Yes	Yes
Product add to cart	Yes	Yes	Yes	Yes	Yes	Yes
PC Builder	Yes	Yes	Yes	Yes	No	Yes
Quick view	Yes	Yes	No	No	Yes	Yes

2.4 Summary

In this part, the available literature was reviewed on the topic of insect pests detection through deep learning algorithms, such as CNNs, Faster R-CNN, and YOLO. It also talked about such fusion models as early and late fusion, and knowledge distillation to enhance classification accuracy and take computational efficiency into consideration. The review revealed some of the gaps in the current literature, which included incomplete coverage of pest species, absence of model generalization, and open-ended scalable solutions to practical cases. Also, it analyzed the application of mobile and web applications to detect pests. The gap analysis indicated that this research will fill these gaps by increasing pest coverage, enhancing model generalization, increasing computational speed and examining model fusion methods.

Chapter 3

Research Methodology

The current chapter describes the research methodology of designing and introducing the pest detection system based on the deep learning methods. It explains how to apply the method of the use of Vision Transformer (ViT) and Data-efficient Image Transformer (DeiT) models, knowledge distillation, data preprocessing, data augmentation and fusion to enhance the accuracy and limit the number of computations when classifying insect pest on a digital image.

3.1 Methodology

Literature on insect pest detection based on deep learning methods such as CNNs, Faster R-CNN, and YOLO, as well as on fusion models (early/late fusion) and knowledge distillation to enhance computational performance and accuracy, were reviewed in this section. It described research gaps including the fact that the range of pest species studied is limited, there is no generalization of models, and the solutions cannot be scaled. Mobile and web applications to detect pests were also reviewed. The areas of this research focus on filling these gaps with increasing the pest coverage, better generalization, faster computation, and model fusion methods.

3.1.1 Overview

The given paper introduces such a model as a deep learning system of insect pest detection founded on ViT and DeiT models. These architectures are optimized around the lesion images and augmented on attention mechanisms (SE, CBAM) to perform feature extraction in a less sensitive way. Knowledge distillation is used, in which knowledge is transferred to a small model, EfficientNetB4 (student), to optimize accuracy and efficiency by ViT (teacher). In fusion techniques (early/late fusion, majority voting) individual into predictions of the various models are fused to form an overall more robust model. The technique takes consideration the processing of various conditions of images by transfer learning, fine-tuned, and data augmentation to provide a fast and accurate diagnosis, especially in areas with few resources

3.1.2 Proposed Methodology

This study uses a high-level deep learning methodology to classify insect pests with a medical image. The dataset is pre-treated (resizing, normalization, and data augmentation) to optimize the model. ViT and Data-efficient Image Transformer (DeiT) are trained on a large-scale dataset and optimized to classify insect pests. The

methodology entails early and late fusion: early fusion integrates the characteristics of the two models whereas late fusion integrates their results and uses majority to improve their accuracy. They apply knowledge distillation with ViT as a teacher model of student model EfficientNetB4. Training of models is done using cross-entropy loss and assessed in accuracy, F1-score and ROC-AUC. The given strategy is designed to provide the effective diagnostic system of pest detection that fits in a wide range of habitats, including the ones of resource-poor ones. The complete workflow is presented in Figure 3.1.

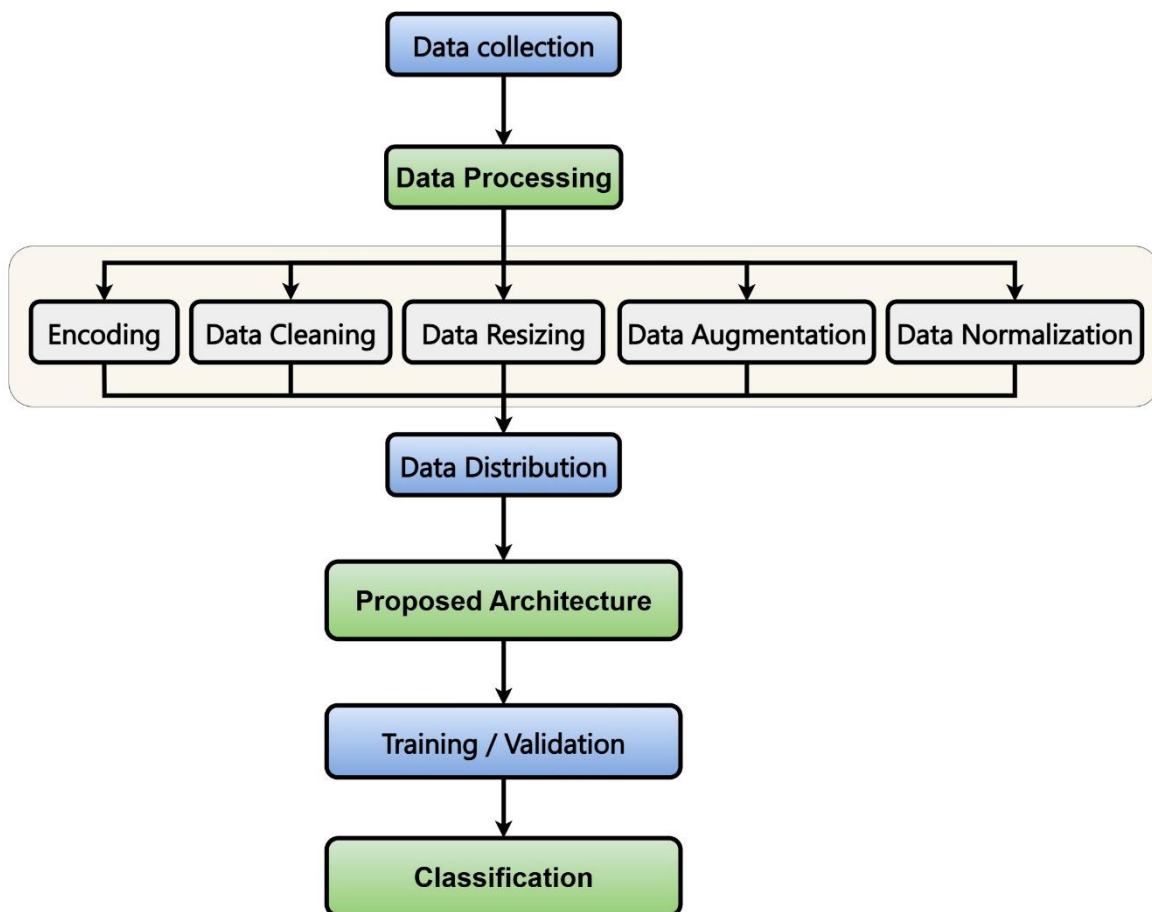


Figure 3.1: Full Workflow

3.2 Detailed Methodology and Design

In the research work, the adopted methodology is dedicated to the creation of a deep learning-based insect pest classification system with the IP102 dataset. The solution is based on pre-trained Vision Transformer (ViT) and Data-efficient Image Transformer (DeiT) models and applies early and late fusion for obtaining a higher degree of accuracy. The majority voting ensembling technique is used to merge the

models' predictions that provides robustness and minimizes bias. ViT teacher model is learned and subsequently knowledge is mapped into the smaller EfficientNetB4 student model to achieve computational efficiency. The data pre-processing techniques presented in the methodology, in their turn, are resizing, normalization, and data augmentation to solve the problem of class imbalance and to provide high-quality training data. The suggested merge of the models and algorithms will likely result in the development of a good, scalable, and accurate pest detection and classification algorithm for real-world applications.

3.2.1 Ensemble Fusion Approaches Using ViT and DeiT for Improved Transfer Learning

In this study, a diagnostic system is suggested that is based on deep learning and its application to early detect insect pests through clinical images, including pest. Conventional diagnostic methods, such as PCR tests demand specific equipment and experience, which may not be accessible in resource-poor environments. Also, insect pests may mimic the clinical presentation of other diseases such as chickenpox and hence they are difficult to detect at an early stage. This has necessitated the development of computerized systems with the ability to differentiate between insect pests and other conditions basing on the images of the pests. Deep learning (DL) and machine learning (ML), and convolutional neural networks (CNNs) in particular have demonstrated achievements in the field of medical image classification and are able to support the detection and tracking of rare diseases such as insect pests on a transfer learning basis.

This paper applies prior trained models such as Vision Transformer (ViT) and Data-efficient Image Transformer (DeiT) and other advanced attention mechanisms such as Squeeze-and-Excitation (SE), Convolutional Block Attention Module (CBAM) to maximize the classification accuracy. The models identify specific image characteristics at high granularity, which is important in the process of differentiating visually similar diseases. ViT and DeiT manipulate pictures as patches in sequence, taking into account global image connections, whereas SE and CBAM blocks enhance feature focus. SE optimizes the classification by refining channel-level attention with recalibration of feature maps, whereas CBAM integrates both channel and spatial attention to prioritize areas of images with important features, thereby improving the classification, particularly in terms of small or subtle lesions. The experiment uses early fusion, in which ViT and DeiT joint outputs are spliced at the feature map level; and late fusion, in which single predictions are pooled and weighted to increase model advantages. An ensemble method involves majority-voting on the summed predictions to minimise bias and variance, producing a more resilient model than either ViT or DeiT alone.

The training incorporates pre-trained models training using transfer learning. ViT and DeiT models originally trained at large scale datasets like ImageNet are fine-tuned on the insect pest data set. The transfer learning strategy would result in minimal data requirements with all the data being labeled yet still have a high level of accuracy. Models are trained with the help of the cross-entropy loss function that are usually applied to multi-class classification. Adam is the optimizer during the training process and is the most appropriate in situations involving sparse gradients and performs efficiently with large models such as ViT and DeiT. The learning rate is low (0.0001) to avoid overfitting and the model converges slowly so that it would not overshoot the optimal parameters.

Among the crucial limitations in implementing such models in the real world is to make the models perform consistently with an image of different qualities and in different conditions. The resolution of the images, the lighting condition in the picture, and the camera quality may be some of the important factors that also play a significant role in the deep learning models working efficiently. As a remedy to this, propose fine-tuning and hyperparameter optimization to enable the models to be improved to address these issues. Fine-tuning means the changing of the features of pre-trained models aimed at making them more suitable for representing the peculiarities of the insect pest dataset, and hyperparameter optimization can help in deciding the most suitable combination of functionalities (parameters) of trained models. In addition, the model performance is evaluated on some variant subsets of the data to check its generalization capacity by different populations and regions thus the model is capable of performing in various real-world situations.

By using the presented methods, the proposed methodology will construct an effective, robust, and efficient diagnostic model of insect pests that may help detect and diagnose insect pest early and save precious time to treatment and prevent the further spread of the virus. Pre-trained models, attention mechanisms, fusion strategies, and ensemble approaches offer a potent paradigm to address the diagnostic issues of insect pests and other diseases similar in nature. The final aim is to develop a system that can be not only clinically valuable but can be implemented in resource-limited scenarios due to time and availability of diagnostics support to curb community transmission of infectious diseases such as insect pests wherever it may occur. Figure 3.3 demonstrates that the suggested methodology of Ensemble Fusion Approaches, based on ViT and DeiT models, allows obtaining better classification outcomes on the aspect of detecting insect pest with use of the advantages of both models by performing early and late fusion.

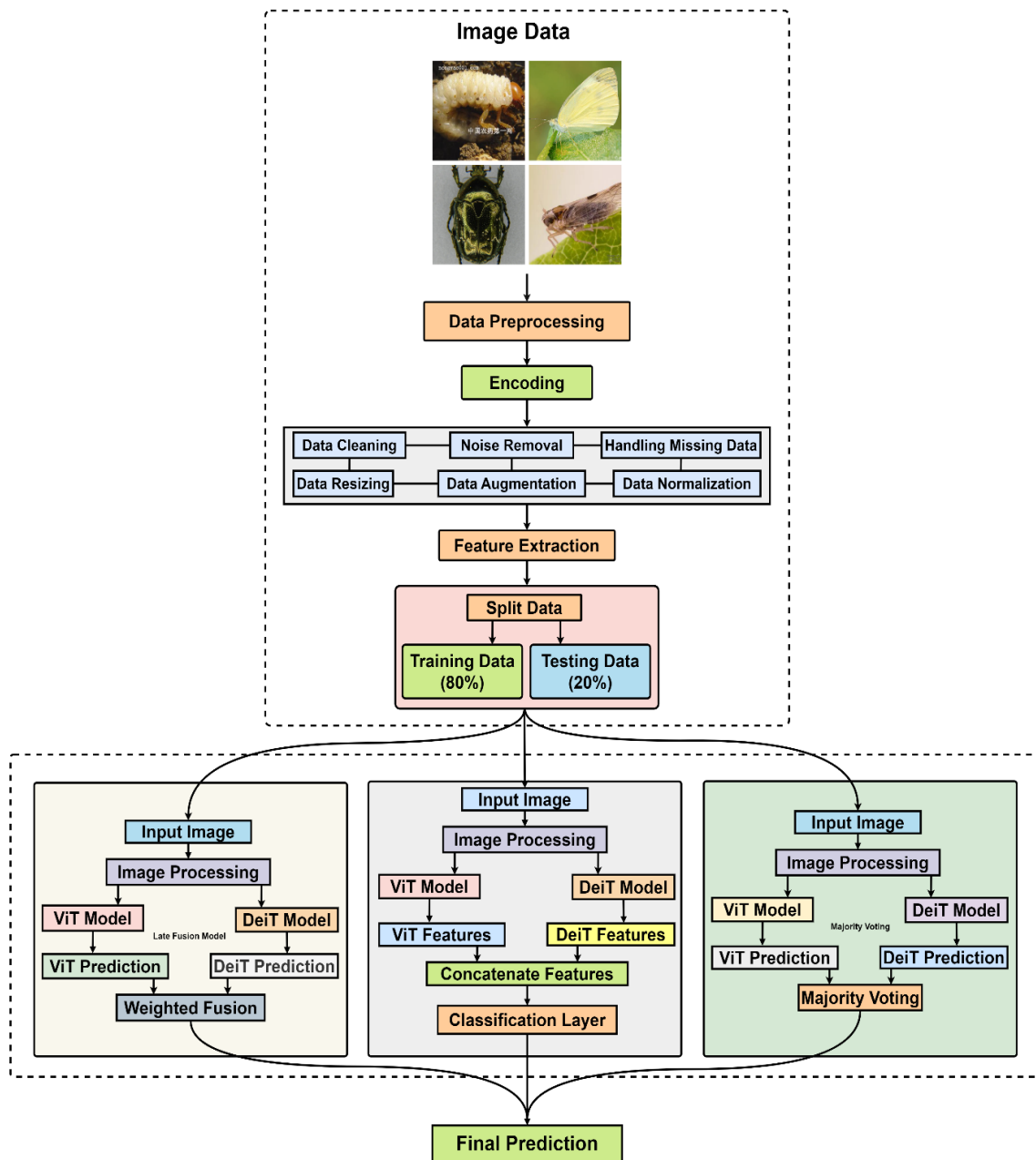


Figure 3.2: Proposed Methodology for Ensemble Fusion Approaches Using ViT and DeiT for Improved Transfer Learning

3.2.2 Optimized Knowledge Distillation for Model Building: A ViT-EfficientNetB4 Hybrid Approach

This paper starts with the collection and preprocessing of data, which formed a dataset of the images of pests caused by various viral diseases (insect pest classes). One of the biggest issues in training deep learning models on rare diseases is the absence of sufficiently labelled images, and as such data augmentation (e.g. rotation, flipping, cropping) is used to mimic real-world conditions and improve the dataset. Normalization of the images is done through ViT image processor and the labels are converted to class IDs. In order to deal with the imbalance in the classes, we take the RandomOverSampler technique by which we can make sure that the model learns all the classes equally.

Vision Transformer (ViT) is taken as the main architecture in model training. ViT model (google/vit-base-patch16-224) is pre-trained on the pests dataset and then fine-tuned on the pest data, transfer learning enhances the accuracy with a smaller number of labeled samples. Besides that, EfficientNetB4 may be applied as a student model in a knowledge distillation approach where it is trained on the ViT teacher model.

The lightweight EfficientNetB4 model that is used in real-time and fine-tuned by a custom classifier is used to achieve efficient and correct pest classification especially in resource-capped settings, such as mobile devices. The distillation process involves computing a combined loss function where the student model learns to minimize both the normal cross-entropy loss function (hard targets) and a soft loss from what the teacher model gives as its output (soft targets).

Therefore, not only does what is available assist in student model learning (the hard targets), but it can also capitalize on what it already knows from what it has gained from the teacher model, which causes it to generalize much earlier. Soft loss is computed from Kullback-Leibler divergence of teacher and student model's predicted distribution and scaled by a temperature parameter to adjust probability distribution smoothness levels.

Distillation's combined loss and hard loss is achieved by a weighted averaged form, where the soft loss to hard loss ratio captures a value of hyperparameter alpha. All models have been trained using the PyTorch library, and models were run by executing within a GPU to be able to have quicker computations reached sooner.

Running of the training occurs over multiple epochs, where either model were updated by Adam optimizer, which automatically adjusts its learning rate. Model performance has been monitored at the end of every epoch by a set of metrics, including accuracy, F1 score, precision, recall, and area under ROC Curve (AUC). All these metrics assist in determining what it is to be able to categorically discriminate against various classes, all the more relevant in a multi-class classification task like ours. Also, tested has been what is termed model robustness by holding a separate set of tests, distinct from the original dataset itself. Overall, this proposed research methodology follows more-complex deep learning techniques like knowledge distillation, transfer learning, and CNN networks to implement effective AI-based detection of insect pests.

Joint usage of ViT and EfficientNetB4, data augmentations, and oversampling methods makes it work well when limited labeled data are available. That knowledge-distillation not only makes the student more efficient but also makes it usable in real low-resource settings and makes it extremely effective to incorporate it within this proposed research methodology.

That pre-trained ones can be fine-tuned, with sophisticated techniques to deal with imbalanced data and data variations, makes it enable accurate, rapid, and affordable detection of insect pests early, thereby eventually leading to its containment within regions affected by insect pests. As was depicted in Figure 3.4, the methodology of Optimized Knowledge Distillation utilizes the Vision Transformer (ViT) as the model teacher and EfficientNetB4 as the model student. This combination of both models gives the advantage of the knowledge shared by the teacher model to the student model to maximize in terms of computational efficiency and accuracy of classification particularly in a resource-constrained setting.

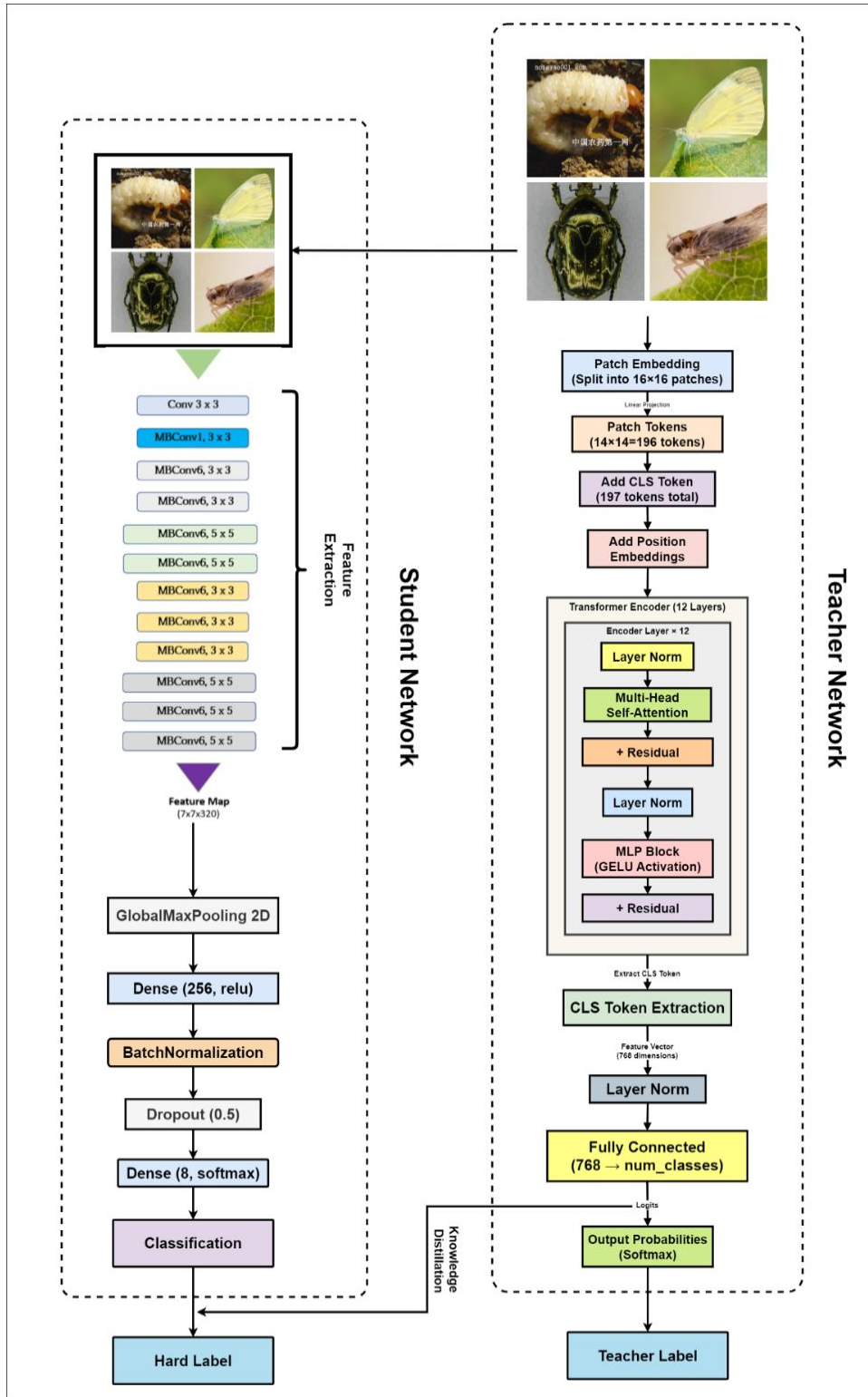


Figure 3.3: Proposed Methodology for Optimized Knowledge Distillation for A ViT-EfficientNetB4 Hybrid Approach

3.2.3 Pre-processing and Dataset Creation

This paper aims at developing a quality dataset of insect pests detection and categorization on images. Preprocessing of the data involves cleaning, resizing, normalization, and augmentation operation (e.g. rotation, flipping, cropping) as indicated in Figure 3.5. The steps train the dataset in the most optimal way, resolving the class imbalance and enhancing model accuracy.

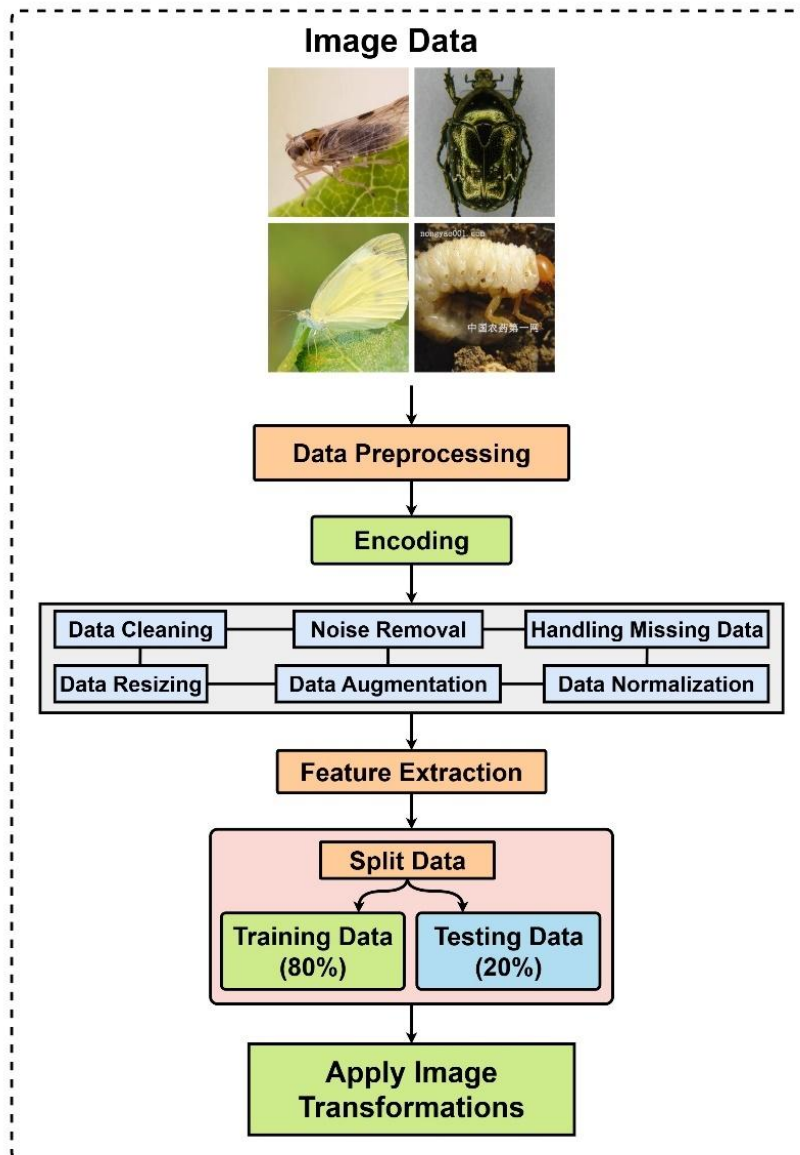


Figure 3.4: Proposed Data Preprocessing

3.2.3.1. Data Collection

The sample of this research will be comprised of pest images as the main symptom of insect pests. These images are obtained publicly available repositories and labeled with different diseases whose symptoms may be similar to insect pests. Since insect pests are rather rare, the data is complemented by the photos of other diseases bearing similar visual characteristics. It has IP102 labels, which contain Grub, *Pieris canidia*, and yellow rice borer. All images differ in the appearance of lesions, light, resolution, and pests types, which enhance more efficient model generalization. Nevertheless, the scarcity of insect pest image provides a challenge on the training of deep learning models. The image classes are those shown in figure 3.5.

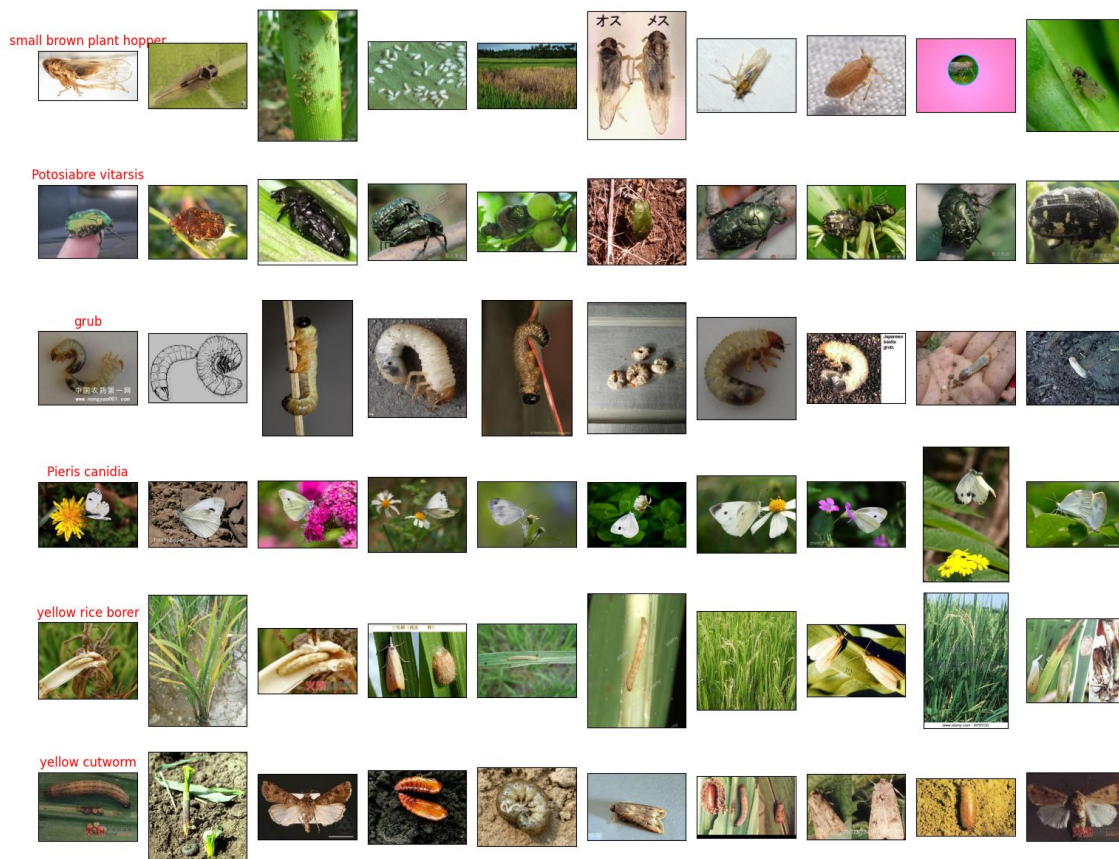


Figure 3.5: All Classes Images

3.2.3.2. Data Pre-processing and Cleaning

The next step in the dataset creation process is pre-processing, which includes data cleaning, resizing, normalization, and augmentation. Proper pre-processing ensures that the images are in a format suitable for deep learning models, thereby improving the efficiency of training and the accuracy of the model.

3.2.3.3 Data Cleaning

Initially, any corrupted or incomplete image files are removed. Since some of the images in the dataset may have been truncated or corrupted during download, utilize the PIL library to ensure that all images are readable. Specifically, set the `ImageFile.LOAD_TRUNCATED_IMAGES = True` flag, which allows the loading of truncated images, ensuring that no valuable data is lost.

3.2.3.4 Resizing and Standardization

Images collected from different sources vary in size, resolution, and aspect ratio. To standardize the input, all images are resized to a fixed dimension, which is compatible with the pre-trained models being used (ViT and DeiT). Both the Vision Transformer (ViT) and Data-efficient Image Transformer (DeiT) models expect inputs of size 224x224 pixels, which is the standard for most pre-trained models. The images are resized to these dimensions while maintaining their aspect ratio, ensuring that key features in the images are preserved.

3.2.3.5 Normalization

Normalization is another crucial pre-processing step, which adjusts the pixel values in the image to a standard range. For the ViT and DeiT models, the pixel values are normalized based on the mean and standard deviation of the pixel values from the ImageNet dataset, which the models were pre-trained on. These models were originally trained with images whose pixel values were normalized with the mean and standard deviation values of [0.485, 0.456, 0.406] and [0.229, 0.224, 0.225] for the three color channels (RGB), respectively. Normalizing the input images using these values ensures that the model's pre-trained weights are leveraged effectively, which improves the overall performance and reduces training time.

3.2.3.6 Dataset Balancing and Augmentation

3.2.3.6.1 Handling Class Imbalance

One of the major challenges in medical image classification tasks is the imbalance in the distribution of samples across different classes. In the case of insect pest, there is a significant class imbalance due to the rarity of the disease and limited available data. To address this issue, employ Random Oversampling using the `RandomOverSampler` from the `imbalanced-learn` library. This method randomly replicates samples from the minority classes, ensuring that all classes have an equal

number of samples during training. This approach is particularly useful for preventing the model from developing a bias towards the majority class (i.e., diseases that are more common in the dataset) and allows the model to learn to distinguish all classes effectively.

3.2.3.6.2 Data Augmentation

To further enhance the model's robustness and improve generalization, data augmentation techniques are applied to the training images. Data augmentation artificially increases the size of the dataset by generating new images through random transformations. These transformations include:

Random Rotation: Rotating the images by random degrees ensures that the model becomes invariant to orientation.

Random Horizontal Flip: Flipping the images horizontally helps the model learn to identify lesions regardless of their orientation.

Random Adjust Sharpness: This transformation randomly adjusts the sharpness of the image, simulating real-world conditions where images may vary in focus.

Random Resized Crop: Cropping and resizing the images ensures that the model learns to focus on different parts of the lesion, rather than memorizing specific regions.

Resize: The images are resized to a fixed size (224x224 pixels) to match the input requirements of the pre-trained models.

These augmentation techniques provide the model with a more diverse set of images, preventing it from overfitting to the limited training data and enabling it to generalize better when deployed in real-world settings.

3.2.3.7 Label Mapping and Dataset Creation

3.2.3.7.1 Label Mapping

Each image is labeled with the corresponding disease class, but deep learning models require numeric labels instead of string labels. To address this, the labels are mapped to numeric values using two dictionaries: `label2id` and `id2label`. The `label2id` dictionary maps the disease names (e.g., insect pest) to unique integer IDs, while `id2label` maps the integer IDs back to the disease names. This mapping is

crucial for training the model, as the model needs to predict numeric values corresponding to each class.

3.2.3.7.2 Dataset Conversion

Once the labels have been mapped to IDs, the dataset is converted into a format that is compatible with the Hugging Face datasets library. The images and labels are converted into a Dataset object, with the images stored as PIL Image objects and the labels as integers. The Dataset class also allows for easy manipulation, such as splitting the data into training and testing sets.

3.2.3.7.3 Training and Testing Split

The dataset is split into training and testing sets using an 80-20 split ratio. This ensures that the model is trained on the majority of the data while still having enough data left for testing and evaluation. The data is stratified by class, ensuring that both the training and testing sets contain a proportional number of samples from each class. This stratification prevents the model from being biased towards certain classes due to uneven representation in the testing set.

3.2.3.8 Final Dataset Overview

After these pre-processing steps, the final dataset consists of 6 classes like *Pieris canidia*, *Potosiabre vitarsis*, grub, small brown plant hopper, yellow cutworm, and yellow rice borer. The dataset is balanced using oversampling, augmented to increase its size and variability, and pre-processed for optimal performance with the ViT and DeiT models. The images are resized, normalized, and transformed, ensuring that the deep learning models can process them effectively. In summary, the pre-processing and dataset creation steps involve data cleaning, resizing, normalization, class balancing, and augmentation. These steps ensure that the dataset is suitable for training robust deep learning models that can accurately classify insect pests from pest images. By using a combination of pre-trained models, data augmentation, and class balancing, this methodology aims to improve the model's generalization capability, ensuring its applicability in real-world clinical settings where the diagnosis of insect pests is critical.

3.2.4 Algorithm Workflow

The algorithm workflow presented in this study involves several key steps that ensure the efficient classification and detection of insect pests from medical images. This process integrates multiple machine learning and deep learning techniques,

including pre-processing, model training, evaluation, and fusion methods, to create an accurate and robust diagnostic tool. The following sections outline the step-by-step workflow used in this study, from data collection and pre-processing to model training, evaluation, and the application of advanced fusion techniques.

3.2.5 Model Architecture

3.2.5.1 Ensemble Fusion Approaches Using ViT and DeiT

ViT and DeiT models are capable of analyzing an image and classifying it by extracting its components and decomposing them. Early fusion takes the representations of the two models and combines them early in life, letting them collaborate to produce a more powerful classification. Late fusion, however, does not modify the individual models itself, but at the final output stage, combines the individual predictions of ViT and DeiT models through logit-weighting. Another form of aggregation is majority voting, where the ultimate decision is taken by the average prediction of the two models, minimizing prediction variance and maximizing the robustness of the models.

3.2.5.1.1 Training and Fine-Tuning

In the paper, the models are also trained with the pre-trained DeiT and ViT architectures, with the initial pre-trained weights fine-tuned on the insect pest dataset to adjust the weights to better classification performance. The training algorithm uses cross-entropy loss, which is appropriate in multi-class classification tasks because it assesses the difference between the correct labels and the probabilities of the prediction in addition to penalizing the wrong guesses. Adam optimizer is used because it is effective in deep learning tasks which take advantage of adaptive learning rates created by gradient changes of the loss function. To tune the hyperparameters, we will use a learning rate of 0.0001, a batch size of 8, and 5 training epochs, which is a good balance between convergence and computation. Then we can also avoid overfitting and improve model generalization through early stopping and model checkpointing, which allows us to use the best-performing model when only a limited number of epochs are trained.

3.2.5.1.2 Ensemble Methods

Ensemble strategies are used to improve the performance of classification, using ViT and DeiT models. Majority voting chooses the most often class prediction according to all models, and early fusion combines the feature representations, then classifies them, and late fusion combines the final logits by a weighted sum. These ensemble approaches combine the strengths of the two models to increase strength

and accuracy, especially with noisy or inconsistent data.

3.2.5.2 Early Fusion Model

Early Fusion Model is a machine learning approach, which merges the features of two or more models at the early phase of the processing and makes a classification of the fused characteristics. The two models (ViT (Vision Transformer) and DeiT (Data-efficient Image Transformer)) in this study find application in the extraction of image features. The point is to merge the features of the two models at the beginning of the pipeline just before the stage of the classification. This enables the model to utilize strengths of the two models in order to enhance accuracy in classification of insect pest lesions on the medical images.

3.2.5.2.1 Mathematical Explanation

Let's begin by breaking down the mathematical formulation for the Early Fusion Model: Feature Extraction from Individual Models:

For any given input image x , the ViT and DeiT models will independently extract features, which can be represented mathematically as:

$$f_{ViT}(x) \in R^{d_1}, \quad f_{DeiT}(x) \in R^{d_2} \dots \dots \dots (i)$$

Here:

$f_{ViT}(x)$ is the feature map extracted by the ViT model for input x ,

$f_{DeiT}(x)$ is the feature map extracted by the DeiT model for input x ,

d_1 and d_2 are the feature dimensions of ViT and DeiT, respectively.

These features represent the internal, high-level representations learned by the two models, and they encode relevant information required for classification.

Concatenation of Features:

The Early Fusion approach involves concatenating the features from the two models. This means that the output from each model, which is a vector of features, is merged into one long vector:

$$f_{fusion} = [f_{ViT}(x); f_{DeiT}(x)] \in R^{d_1+d_2} \dots \dots \dots (ii)$$

Where:

$[\cdot; \cdot]$ denotes the concatenation operation along the feature dimension,

f_{fusion} is the concatenated feature vector combining both feature maps.

This combined feature vector now holds information from both models and will be passed to the next stages of the network, where it will be processed for classification.

Fusion Layer:

Once the features are concatenated, they are passed through a fusion layer, which is a fully connected layer that reduces the dimensionality and learns to combine the features from both models. The fusion layer can be mathematically represented as:

$$f_{fused} = W_{fusion} \cdot f_{fusion} + b_{fusion} \dots \dots \dots (iii)$$

Where:

$W_{fusion} \in R^{512 \times (d_1 + d_2)}$ is the weight matrix for the fusion layer,

$b_{fusion} \in R^{512}$ is the bias term,

$f_{fused} \in R^{512}$ is the output of the fusion layer.

Here, the fusion layer is learned during training to find the most informative features from the concatenated vector. It reduces the dimensionality of the combined features to 512, ensuring that the model focuses on the most relevant parts of the feature space.

Classifier:

After the fusion layer, the resulting fused features f_{fused} are passed through a classifier. The classifier is a simple fully connected layer that outputs the logits for each class. The logits can be mathematically expressed as:

$$\hat{y} = W_{classifier} \cdot f_{fused} + b_{classifier} \dots \dots \dots (iv)$$

Where:

$W_{classifier} \in R^{C \times 512}$ is the weight matrix for the classifier,

$b_{classifier} \in R^C$ is the bias term of the classifier,

$\hat{y} \in R^C$ is the output logits, where C is the number of classes.

The softmax function is applied to the logits to convert them into class probabilities:

$$\hat{y}_{pred} = softmax(\hat{y})$$

Where the softmax function is:

$$softmax(\hat{y})_i = \frac{e^{\hat{y}_i}}{\sum_j e^{\hat{y}_j}} \dots \dots \dots (v)$$

This will give the predicted class probabilities, and the class with the highest probability is chosen as the model's prediction.

Loss Function:

The Early Fusion model is trained using the cross-entropy loss function, which is commonly used for classification problems. The cross-entropy loss for multi-class classification is given by:

$$L(\hat{y}, y) = - \sum_{i=1}^C y_i \log(\hat{y}_i) \dots \dots \dots (vi)$$

Where:

$\hat{y}_i$ is the predicted probability for class i ,

y_i is the ground truth for class i (1 if the sample belongs to class i , 0 otherwise),

C is the number of classes.

The EarlyFusionModel class takes the ViT and DeiT models, extracts the features using their respective feature extraction layers, concatenates the features, passes them through a fusion layer, and then performs classification. The model is trained using the cross-entropy loss, and Adam optimizer is employed for parameter optimization during training.

3.2.5.3 Late Fusion Model

The Late Fusion Model is an ensemble learning strategy that involves simply averaging the predictions (classification decisions) of two or more models, once they individually complete the classification task. The two models being applied in this study are the ViT and the DeiT which are individually used to classify the input images and their output (logits) is combined via weighted summation. This has the merit that every model brings its own set of advantages to the overall decision and enhances the robustness and accuracy of the classification.

3.2.5.3.1 Mathematical Explanation

The Late Fusion model works by aggregating the output of the individual models after both models have made their predictions. The concept for this model is to take the logits from each model, combining those logits as a weighted sum, and conducting the final classification. Feature Extraction and Prediction:

Let $f_{ViT}(x)$ and $f_{DeiT}(x)$ represent the logits produced by the ViT and DeiT models for an input x . These logits are the raw class scores for each class:

$$f_{ViT}(x) \in R^C, \quad f_{DeiT}(x) \in R^C \dots \dots \dots (vii)$$

Where C is the number of classes in the classification task.

Weighted Sum of Logits:

In Late Fusion, the logits from both models are combined using a weighted sum. Let w_{ViT} and w_{DeiT} be the weights assigned to the logits from the ViT and DeiT models, respectively. The combined logits are calculated as:

$$f_{combined} = w_{ViT} \cdot f_{ViT}(x) + w_{DeiT} \cdot f_{DeiT}(x) \dots \dots \dots (viii)$$

Where w_{ViT} and w_{DeiT} are learnable parameters that determine the contribution of each model to the final prediction.

Final Classification:

The combined logits are passed through a softmax function to convert them into class probabilities:

$$\hat{y}_{combined} = softmax(f_{combined})$$

Where the softmax function is defined as:

$$\text{softmax}(f_{combined})_i = \frac{e^{f_{combined,i}}}{\sum_j e^{f_{combined,j}}} \dots \dots \dots (xi)$$

This results in the predicted probabilities for each class, and the class with the highest probability is selected as the final prediction.

In the LateFusionModel class, the logits from the ViT and DeiT models are computed independently. These logits are then weighted and summed to form the combined logits, and the softmax function is applied to these logits to get the final class probabilities. This process allows the model to use the strengths of both ViT and DeiT in the decision-making process.

3.2.5.4 Ensemble Methods and Voting: Majority Voting

The ensemble methods form a group of machine learning techniques where several models are combined in order to benefit performance. The rationale here is that through the integration of the outputs of multiple models it lessens the variance of the prediction and better generalizes. A common and simple but powerful ensemble technique is Majority Voting where the predictions of several models are joined by performing a majority vote on the classes where the class had the greatest number of predictions among the combined models.

3.2.5.4.1 Mathematical Explanation

In Majority Voting, each model makes a prediction, and the final decision is based on the majority of those predictions. This method can be formalized mathematically as follows:

Predictions:

Let $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N$ represent the predictions from N different models for a given input x , where each $\hat{y}_i$ is the predicted class for that model:

$$\hat{y}_i \in \{1, 2, \dots, C\}, \quad i = 1, 2, \dots, N \dots \dots \dots (x)$$

Where C is the number of classes, and N is the number of models in the ensemble.

Majority Voting:

The final prediction $\hat{y}_{final}$ is determined by selecting the class that is predicted most frequently by the models. For a binary classification problem, the Majority Voting decision can be represented as:

$$\hat{y}_{final} = \text{mode}(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N) \dots \dots \dots (xi)$$

Where $\text{mode}(\cdot)$ represents the mode (the most frequent value) of the predictions.

For multi-class classification, Majority Voting can be extended by using the mode function across all models. This approach helps in ensuring that the final decision is robust and reflects the consensus of the ensemble. In this phase, the majority vote function calculates the final prediction by aggregating the predictions from both the

ViT and DeiT models. The predictions are combined, and the final decision is made based on the majority of votes.

3.2.5.5 Innovation and Novelty in Model Design

The study presents a hybrid model, that is, a combination of two of the most recent transformer-based models: Vision Transformer (ViT) and Data-efficient Image Transformer (DeiT), that employs two methods of early and late fusion with a majority voting ensemble. Such a multi-model fusion enables the model to exploit the features of the two architectures: ViT is better-adapted to capture the long interactions, whereas DeiT is more geared towards real-time applications. Early fusion combines the attributes of the two models at the initial stages and late fusion combines the outcomes of the two models at the advanced stages without sacrificing the identity of each model. A majority voting increases robustness through the aggregation of model outputs. Such fusing and ensemble approach enhances accuracy, generalization, and real-time performance especially in the environments that are constrained in resources. The small and skewed data capability of the model and its real-time prediction capabilities on mobile-friendly architectures are a strong improvement over insect pest disease classification, and can serve as a viable solution to disease detection on low-resource conditions.

3.2.5.6 Experiment

3.2.5.6.1 Vision Transformer (ViT)

Vision transformer (ViT) A transformer model-based image classification inspired by transformer models in natural language processing (NLP). In contrast to the old fashioned CNNs, which make pixel-wise decisions, ViT divides an image into patches and considers each patch as a token. These tokens are fed through transformer layers which learn the long-range dependencies between patches to allow ViT to learn global image patterns. ViT needs huge datasets to be pre-trained but has an outstanding result on complex image classification. It has demonstrated competitive performance with CNNs over large data sets such as ImageNet. This paper implements ViT in an ensemble framework to retrieve features in image patches, and uses its global context knowledge to classify the disease correctly.

3.2.5.6.2 Data Efficient Image Transformer (DeiT)

Data-efficient Image Transformer (DeiT) is another variant of ViT that was created to diagnose the large-scale data requirement in pre-training. A smaller student model (DeiT), with a distillation-based strategy, emulates the performance of a

larger teacher model (typically ViT), allowing high accuracy with fewer data. Similar to ViT, DeiT also uses transformer layers to process image patches, and is more efficient because of its distillation method hence suitable when there are limited datasets or high-training expense. DeiT, in this work, is added to the ensemble model, the predictions of which are combined with those of ViT to increase the accuracy of the classification. Both ViT and DeiT are transformer-based networks but do not have the same training requirements. ViT is a good model with large datasets, that captures the complex image relationship, but requires substantial data and computational resources. DeiT, in its turn is more efficient and effective when working with smaller datasets, due to knowledge distillation. The system is a hybrid of the two systems as ViT achieves high performance due to learning in global context, whereas DeiT is efficient and accurate with smaller datasets. The fused model as illustrated in Figure 3.7 results in the combination of the benefits of the two models through early and late fusion strategies to enhance insect pest classification precision.

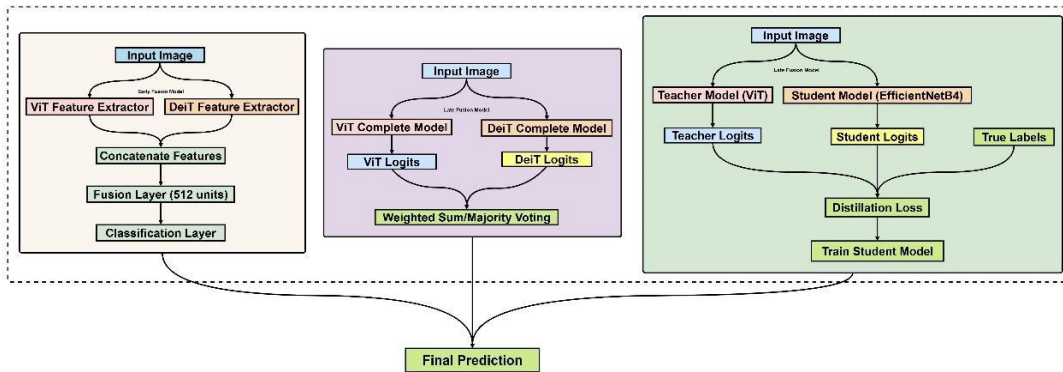


Figure 3.6: Proposed Architecture of Ensemble Fusion Approaches Using ViT and DeiT

3.2.5.7 Optimized Knowledge Distillation for Model Building: A ViT-EfficientNetB4 Hybrid Approach

The algorithm is centered in the area of knowledge distillation where a smaller model (student model) that is efficient is trained on the expertise of a much more comprehensive and pre-trained model (teacher model). The teacher model is the ViT whereas the student model is the EfficientNetB4. The insect pest dataset is fine tuned after the pre-trained model on a large dataset (ImageNet). The student model is trained based on the ground truth labels as well as soft outputs of the teacher model. This unites the soft loss (relying on the predictions that the teacher used) with the hard loss (the cross-entropy loss), as predicted by the student model. The student model could simulate the behavior of the teacher model and therefore

surpass the performance of the efficient model- even when the size of the computational complexity is cut down.

3.2.5.7.1. Training and Fine-Tuning

TheViT teacher model and EfficientNetB4 student model are trained/fine-tuned on the insect pest dataset. As part of the model training, the weight was set to initialisation of the pre-trained weights of the two models then the insect pests of the training set were fine-tuned. The training loss was cross-entropy that represents the distinction between the label prediction and real label. In every model there are weights reps of the finest dimensions of the model in order to attain loss. In this case Adam used an optimizer to update or alter the representation of the model and is using a small value of 0.0001 as it is suitable in fine tuning models. It has repetitions or epochs of training (5 in this case) that matter to give the models an opportunity to grasp meaningful features of the lesions When are training using the student model are training it by using knowledge distillation whereby, the student model would be learning given the real labels (hard targets) and by the soft predictions of the teacher model on training sample which are (soft targets) during the training epoch of the teacher model. The loss as distilled in this case is a weighted average between the hard loss and the soft loss which enables the student model to not just parrot the teacher model but to also enable it to generalise the teacher model well on to the new dataset.

3.2.5.7.2 Model Evaluation

After training, the model is evaluated using a range of performance metrics to assess its classification accuracy. These metrics include:

Accuracy: The overall proportion of correctly classified images compared to the total number of images in the test set.

F1-Score: The F1-score is calculated to balance precision and recall, which is crucial for ensuring the model does not overpredict or underpredict any class.

Confusion Matrix: This matrix provides detailed insights into how well the model performs on each class, showing the number of correct and incorrect predictions for each class.

ROC AUC: The ROC curve is plotted, and the AUC (Area Under the Curve) score is calculated to measure the model's ability to distinguish between different classes across varying thresholds.

These metrics help evaluate the model's performance on the test set and provide insights into areas for improvement.

3.2.5.7.3 Knowledge Distillation Loss

The knowledge distillation process is a key aspect of this workflow. The teacher model (ViT) provides soft predictions, which are used to guide the student model (EfficientNetB4) during training. The distillation loss function combines the hard loss (cross-entropy loss) with the soft loss (KL divergence) to help the student model mimic the teacher model's behavior. This process allows the student model to learn from the teacher model's predictions, making it more accurate and efficient despite its smaller size. The distillation loss has two key components:

Hard Target Loss: This is the traditional loss function that compares the model's predictions to the true labels.

Soft Target Loss: This loss is based on the teacher model's predictions and encourages the student model to match the teacher's output distribution, thus transferring knowledge from the teacher to the student.

The balance between the hard and soft loss is controlled by a hyperparameter, which is optimized during training.

3.2.5.7.4 Final Evaluation and Comparison

After training and evaluating the teacher and student models, the results are compared to determine which model performs best. The comparison is based on key metrics such as accuracy, F1-score, and AUC. The final decision on which model to deploy for real-world use is based on these evaluation results. The best-performing model is then selected for deployment, providing an efficient and reliable tool for insect pest detection in clinical settings. The algorithm workflow described in this study leverages knowledge distillation to create an efficient student model that can accurately classify insect pests. By using a pre-trained ViT model as the teacher and fine-tuning an EfficientNetB4 model as the student, this approach balances performance and efficiency. The use of distillation loss allows the student model to learn from the teacher's soft predictions, improving its classification accuracy. The final model, after training and evaluation, offers a robust solution for the early detection of insect pests, making it suitable for real-world applications, particularly in resource-constrained environments.

3.2.5.7.5 ViT (Vision Transformer) Model

Vision Transformer (ViT) is a model, which conceptualizes the ways in which image information can be transformed to incorporate transformer-based architectures, which are originally used in Natural Language Processing (NLP). Instead of working on local patches of the image as conventional Convolutional Neural Networks (CNNs), the ViT model takes an image as a series of patches, e.g., as words are staggered into tokens in NLP tasks. This enables the ViT to learn over long-range dependencies of images which is useful where the global context is required.

3.2.5.7.5.1 Mathematical Explanation

Let's break down the mathematics of ViT. The process starts with the input image $x \in R^{H \times W \times C}$, where H is the height of the image, W is the width, and C is the number of channels (typically 3 for RGB images). The first step is to split the image into non-overlapping patches.

Patch Division and Flattening:

The image is divided into $P \times P$ patches. Let p_i denote the i-th patch, where each patch p_i has a dimension of $P \times P \times C$. After dividing, flatten each patch into a vector of size $P^2 \times C$. The number of patches is $N = \frac{H \times W}{P^2}$, where N is the total number of patches. Thus, the flattened patches are represented as:

$$x_{patches} = [p_1, p_2, \dots, p_N] \quad \text{where} \quad p_i \in R^{P^2 \times C} \dots \dots \dots (xii)$$

These patches are then linearly embedded into vectors of a fixed size d , called the embedding dimension:

$$z_i = W_{embed} p_i + b_{embed} \quad \text{where} \quad z_i \in R^d \dots \dots \dots (xiii)$$

Here, $W_{embed} \in R^{d \times P^2 \times C}$ is the embedding matrix, and $b_{embed} \in R^d$ is the bias term.

Positional Encoding:

Since transformers do not have a built-in notion of the spatial order of patches, positional encodings are added to the patch embeddings to inject information about the relative positions of the patches:

$$z_i^{pos} = z_i + PE_i \dots \dots \dots (xiv)$$

Where PE_i is the positional encoding for the i-th patch, which can be learned or fixed, and z_i^{pos} represents the positional encoding-enhanced patch embedding.

Transformer Layers:

The patch embeddings are fed into a series of transformer blocks. A transformer block consists of a multi-head self-attention mechanism and a position-wise

feedforward neural network (FFN). The self-attention mechanism computes the attention weights between all patches:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \dots \dots \dots (xv)$$

where Q, K, and V are the query, key, and value matrices, respectively, and d_k is the dimensionality of the key vectors. The attention operation allows the model to learn the relationships between all pairs of patches in the image. This process is repeated across multiple layers.

Classification:

The final output from the transformer is the representation corresponding to the special [CLS] token (used for classification), which is passed through a fully connected layer to produce the logits:

$$\hat{y}_{ViT} = W_{cls}z_{CLS} + b_{cls} \dots \dots \dots (xvi)$$

where z_{CLS} is the embedding of the [CLS] token, and W_{cls} and b_{cls} are the classifier's weights and biases. The softmax function is applied to obtain class probabilities:

$$y_{pred} = softmax(\hat{y}_{ViT}) \dots \dots \dots (xvii)$$

Where:

$$softmax(\hat{y}_{ViT})_i = \frac{e^{\hat{y}_{ViT,i}}}{\sum_j e^{\hat{y}_{ViT,j}}} \dots \dots \dots (xviii)$$

This final output y_{pred} represents the predicted probabilities for each class.

Teacher Model: Fine-Tuned ViT

The fine-tuned ViT model is trained on a specific dataset, which in this case is the insect pest dataset. Transfer learning is applied to the ViT model, where a pre-trained model is further optimized to perform well on the specific task. This is achieved by freezing the lower layers of the ViT model and fine-tuning the final layers.

The fine-tuning process involves updating the weights of the final layers to minimize the classification loss, typically cross-entropy loss:

$$L_{CE} = - \sum_{i=1}^C y_i \log(\hat{y}_i) \dots \dots \dots (xix)$$

Where C is the number of classes, y_i is the true label (one-hot encoded), and $\hat{y}_i$ is the predicted probability for class i .

The teacher model is then used to generate soft targets for knowledge distillation, where the logits produced by the teacher serve as the guidance for the student model

during training.

3.2.5.7.6 EfficientNetB4 (Student Model)

EfficientNetB4 is a deep convolutional neural network (CNN) belonging to the family, EfficientNet known to show state-of-the-art accuracy with efficiency in terms of computational resources utilized. The main innovation of EfficientNet is compound scaling, i.e., balance in scaling the depth, width and input resolution of the network. This method optimizes the performance, without making the model too complex, which makes this method appropriate to be implemented in those resource-limited settings, including mobile devices or edge devices. EfficientNet scales the model with a compound coefficient that is used to scale depth, width, and resolution of the network:

$$Depth = \alpha^d, \quad Width = \beta^w, \quad Resolution = \gamma^r \dots \dots \dots (xx)$$

Where α , β , and γ are scaling coefficients, and d , w , and r are the base values for depth, width, and resolution, respectively. The scaling ensures that the model is balanced and optimized for both accuracy and computational efficiency. This scaling approach can be formally represented as:

$$Efficiency = \frac{Accuracy}{Computational\ Cost} \dots \dots \dots (xxi)$$

Where the computational cost is related to the number of parameters in the model, and accuracy refers to the model's performance on the validation set. The EfficientNetB4 model achieves higher accuracy with fewer parameters by carefully balancing these factors.

The architecture of EfficientNetB4 includes depthwise separable convolutions, which separate the spatial and channel-wise convolutions. This reduces the number of parameters and computations compared to standard convolutions. Mathematically, the depthwise separable convolution can be expressed as two operations: depthwise convolution and pointwise convolution.

Depthwise Convolution: The depthwise convolution operates on each input channel separately:

$$y = W_{depth} * x \dots \dots \dots (xxii)$$

Where x is the input feature map, W_{depth} is the filter, and $*$ denotes the convolution operation.

Pointwise Convolution: After the depthwise convolution, a pointwise convolution (a 1x1 convolution) is applied to combine the output from the depthwise convolution:

$$y = W_{pointwise} * x \dots \dots \dots (xxiii)$$

Here, $W_{pointwise}$ is the 1x1 filter used to combine the outputs of the depthwise convolution.

The output from the depthwise separable convolution is a more efficient representation of the input, which helps reduce the computational cost of the model while maintaining accuracy.

EfficientNetB4 is used as the student model in this research. The student model is trained using the teacher's soft targets via knowledge distillation, which allows it to learn from the teacher's knowledge. The student model learns to approximate the teacher's output while being smaller and more efficient in terms of computational resources.

The distillation loss function for EfficientNetB4 as the student model is given by:

$$L_{distill} = \alpha L_{soft} + (1 - \alpha) L_{hard} \dots \dots \dots (xiv)$$

Where L_{soft} is the KL divergence between the teacher's output and the student's output, and L_{hard} is the cross-entropy loss between the student's predictions and the true labels. The parameter α controls the weight between the soft and hard losses.

EfficientNetB4 as the student model benefits from the knowledge distillation process, learning the complex representations of the ViT teacher model. Through the distillation process, the student model retains much of the performance of the teacher while being computationally efficient, making it ideal for deployment on resource-limited devices.

3.2.5.7.7 Innovative Approach and Model Contribution

The novelty of the present work is the integration of the Vision Transformer (ViT) with EfficientNetB4, with knowledge distillation. Here the larger and more complicated ViT model (teacher) passes knowledge on to the smaller and more efficient EfficientNetB4 model (student), which can also gain similar accuracy with fewer computational demands. EfficientNetB4 employs depthwise separable convolutions to reduce the computational expenses without affecting its performance. Both the soft targets (teacher-imposed) and the hard targets (true labels) are advantageous to the student model to improve learning without compromising generalization. The approach enables high precision at low computational expenses and thus is suitable in resource-constrained devices such as mobile devices. This knowledge distillation is optimized as shown in Figure 3.7.

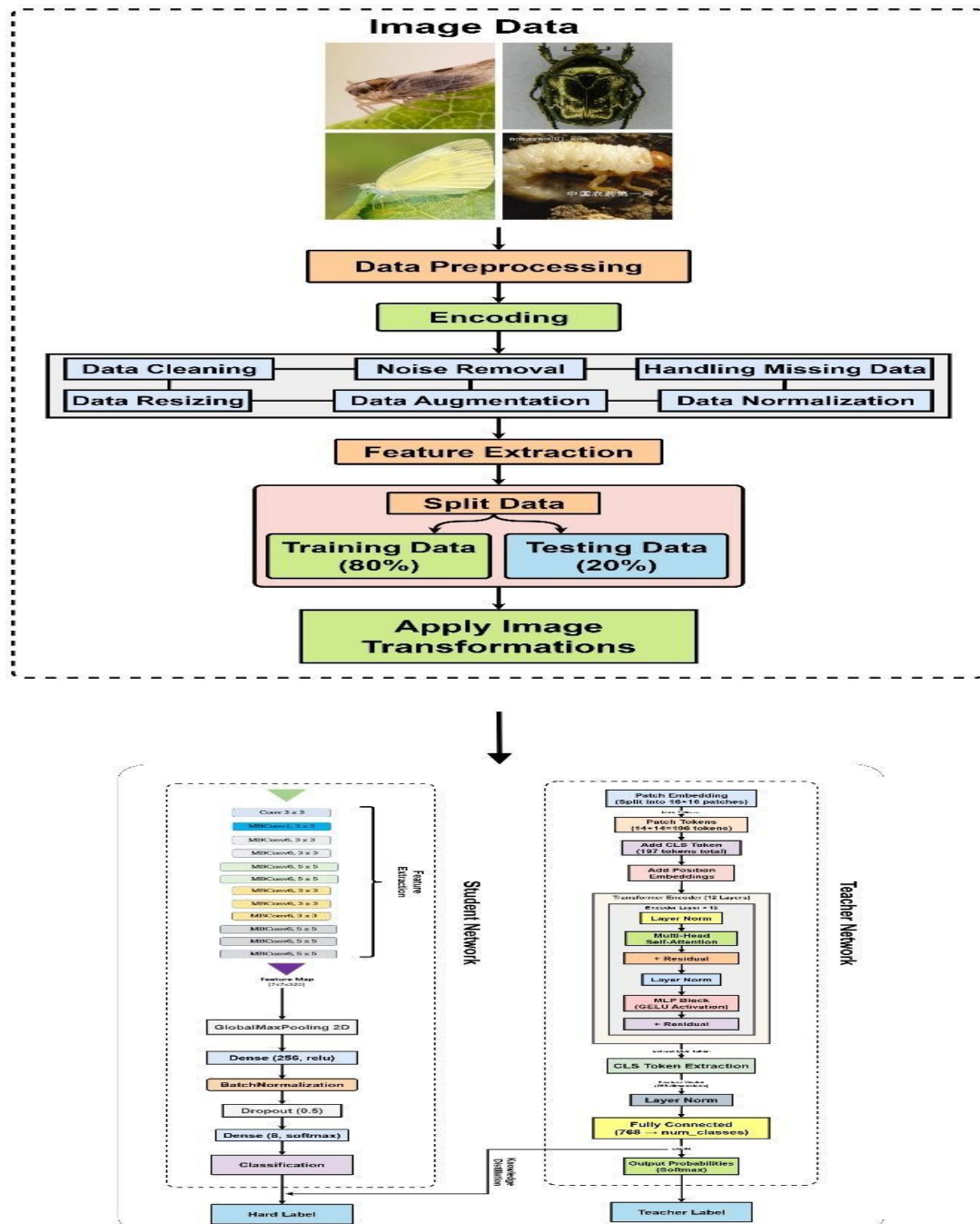


Figure 3.7: Optimized Knowledge Distillation Architecture for Model Building: A ViT-EfficientNetB4 Hybrid Approach

3.3 Project Plan

Insect Pests Classification by Deep Learning Insect Pests are married using deep learning. The proposed study will deliver an identified insect pest detection system that will be supported by a deep learning approach to classify six types of insect pests

presented in an IP102 dataset.

The methodology of the research is based on deep learning and the main steps are the following ones:

Data Collection, Pre-processing: The dataset used is IP102 that will be used in training and testing. The preprocessing process will involve resizing of images, their normalization, and their data augmentation to increase diversity and quality of the data set. The oversampling techniques will be applied to balance the class distribution to overcome the biasedness of pest classes.

Model Design: The main model to be utilized will be the feature extractor model that will consist of the Vision Transformer (ViT), and Data-efficient Image Transformer (DeiT). Such transformer models will be fine-tuned using IP102 dataset to focus on minute details on pests.

Fusion Techniques: The techniques of combining the models will employ early fusion and late fusion techniques in the enhancement of the accurateness of classification. early fusion: the two model feature maps are fused, late fusion: the output of both models ready to make a prediction is fused. Also, using a majority voting method is going to be implemented to extend the decision-making process.

Knowledge Distillation: Knowledge distillation will be used in order to enhance efficiency of the models. ViT will be used as the teacher model and EfficientNet B4 as the student model. The distillation process will be applied such that it relocates learning knowledge between the teacher and student model and as a result, the student model will be computationally cost-effective and possess high classification accuracy levels.

Model Evaluation: The evaluation or performance of the models will be based on accuracy, F1-score, precision, recall, and AUC. Cross-validation will be utilized in order to determine the generalization capacity of the model when used on diverse divisions of the dataset.

3.3.1 Project Plan Phase

This research will be carried out in multiple phases over the course of eight months:

Phase 1: Literature Review & Research Setup:

- Study existing methods for pest detection using deep learning.
- Understand the IP102 dataset and identify gaps in existing methods.

Phase 2: Data Collection & Preprocessing:

- Gather and preprocess the dataset, including data augmentation and class balancing techniques.

Phase 3: Model Development & Experimentation:

- Implement and fine-tune ViT and DeiT models.
- Experiment with early and late fusion methods.

Phase 4: Knowledge Distillation & Optimization:

- Implement knowledge distillation between ViT (teacher) and EfficientNet B4 (student).
- Fine-tune the models with hyperparameter optimization.

Phase 5: Model Evaluation :

- Evaluate model performance using various metrics.
- Perform cross-validation and adjust hyperparameters for improvement.

Phase 6: Conclusion & Report Writing:

- Analyze the results and prepare a detailed report documenting the research findings.
- Write and submit the final research paper.

3.3.2 Research Resources

- **Hardware:** High-performance GPU (e.g., NVIDIA RTX 3080 Ti) for training deep learning models.
- **Software:** Python programming language with libraries such as TensorFlow, PyTorch, scikit-learn, and Keras for implementing models and training.
- **Dataset:** IP102 dataset containing images of 102 insect pest species.
- **Tools:** GitHub for version control, Jupyter Notebooks for model development, and Google Colab for cloud-based training.

3.3.3 Risk Management

- **Data Scarcity:** Augment the dataset using techniques like rotation, flipping, and cropping to artificially increase the dataset size.
- **Class Imbalance:** Use oversampling techniques such as Random Oversampler to ensure equal representation of all classes.
- **Overfitting:** Apply early stopping, regularization, and cross-validation during model training to avoid overfitting.

3.3.4 Expected Deliverables

- **Deep Learning-Based Pest Classification System:** A robust system capable of classifying insect pests from images using deep learning techniques.
- **Research Paper:** A detailed paper documenting the methodology, experiments, and results.
- **Codebase:** Well-documented code for model implementation, training, and testing. The deep learning models that will be used in this research will contribute greatly in detecting pests in the agricultural fields thus transforming the pest classification process into a more accurate and efficient one. A possible solution to the problem of detecting and classifying insect pests in agricultural environments and decreasing the destruction of crops can be the combination of the ViT transfer learning method, DeiT method, fusion mechanism, and knowledge distillation procedures to solve the problem. As a result of this study, one can achieve scalable pest detection systems that can be deployed in resource-constrained environments and eventually benefit the global agricultural community.

3.4 Task Allocation

This table depicts the timeline of the principal activities in each period of the project, from week 12 to week 48.

Table 3.1: Principal Activities in each period of the project

Tasks	Weeks																		
	1 2	1 4	1 6	1 8	2 0	2 2	2 4	2 6	2 8	3 0	3 2	3 4	3 6	3 8	4 0	4 2	4 4	4 6	4 8
Data collection phase																			
Preprocess all the data																			
Model training																			
Create a demo application.																			

3.5 Summary

In this chapter, the authors describe the research methodology of the proposed study on the classification of insect pests based on advanced models of deep learning. The research methodology involves the incorporation of attention mechanism including Squeeze-and-Excitation (SE) and Convolutional Block Attention Module (CBAM), which will be applied to extract the features, through applying model based on the Vision Transformer (ViT) and Data-efficient Image Transformer (DeiT). Further, the knowledge distillation method is used, with the ViT model (teacher) transferring its knowledge to a smaller EfficientNetB4 model (student), without reducing the accuracy, in order to achieve a computationally efficient model. There are a number of fusion techniques integrated into the system, including the early fusion, late fusion, and the majority vote to provide robustness to the predictions. In the training part, the pre-trained models will be refined on insect pest dataset via the use of transfer learning, cross-entropy loss and Adam optimizer. Data preprocessing measures such as resizing, normalization, and data augmentations to provide a top-quality input are also involved in the methodology. This method will develop a scale-able, accurate and effective diagnostic tool that can be applied in a real-time clinical application especially in underserved contexts. The chapter also indicates the stages involved in the project, materials needed together with strategies to be applied in managing risks in order to make the study successful.

Chapter 4

Implementation and Results

The chapter gives an overview of how the system is implemented, the setting up of the environment, the training of the model, and the methodology of how the evaluation processes will take place within this study. The results of the experiments and evaluation of the experimental performance calculated against the various models like Vision Transformer (ViT), Data-efficient Image Transformer (DeiT), and different fusion methods are also given together with the discussion of the attained performance metrics.

4.1 Environment Setup

4.1.1 Hardware Requirements

The hardware configuration of this study is imperative to effective processing and training of the models in the context of high computational complexities of deep learning models. It requires a high-performance GPU which is the kind used in NVIDIA RTX 3080 Ti or others making the training faster to process the large-scale image dataset and complex models like the Vision Transformer (ViT) and Data-efficient Image Transformer (DeiT). Transformer-based models and most forms of deep learning need excessive computing to utilize the intensifying loads of data and advanced computations. Moreover, it will be necessary to have enough storage (at least 100GB) to save IP102 dataset, model checkpoints and the intermediate outputs that will be produced during training and evaluation phases.

4.1.2 Software Requirements

In this study, the system preparation will be performed on Windows, Linux or macOS, yet Linux (Ubuntu) is the most popular operating system as it is much more compatible with machine learning libraries and GPU acceleration. Python is the main programming language to write the library and the implementation of the models, carry out the preprocessing of the data and train the models. In the arrangement, the installation of CUDA and cuDNN will be used to accelerate GPUs. Such tools are required to accelerate training of models, in cases where there are very large data points and complicated computations.

4.1.3 Frameworks and Libraries

To conduct the research, the researchers employ several deep learning frameworks, the main one being TensorFlow. In TensorFlow, it is easy to build and train very complicated models of learning. It is coupled with Keras, a high level API that makes building and

training a model easier. Also, another deep learning framework is PyTorch, which is applied to the implementation of such key techniques as knowledge distillation, which is the main element of the suggested model. Such frameworks together with NumPy, Pandas, Matplotlib data setups to handle and visualize data are at the heart of the research environment. Also, the model testing and pre-processing are performed by using the scikit-learn library.

4.1.4 Dataset Handling and Preprocessing Tools

The largest dataset used in the present research is IP102 dataset that consists of 102 pest species. It has referred to photos of insect pests that are pre-processed in the form of libraries such as PIL (Python Imaging Library) to adjust the size and format pictures into deep learning models. Data augmentation, done to boost the number of data, is also achieved with the ImageDataGenerator module in Keras or albumentations package. Rotations, flipping and the cropping performed with the aim to artificially increase the size of a data are among such augmentations that promise to improve the model ability to generalize to the real-world conditions. Label mapping occurs in a similar manner to digitalize the labels of the pest species in order to trick them to be easily handled by the machine learning algorithms.

4.1.5 Version Control and Collaboration

For reproducibility and the ability to keep a history of the project that can be accessed, the project version control is kept using Git so that changes can be tracked within the codebase. The project code is stored on a GitHub leading to a code repository hub to glean information on sharing and collaboration of code. This will allow team members to work together smoothly and offer variations of the model and datasets during the process of research.

4.1.6 Model Training Configuration

The training set-up will involve loading pre-trained models viz. ViT and DeiT, which have been pre-trained on large data sets, e.g., ImageNet. The training of the models using the IP102 dataset qualifies the models as specialist models in the detection of insect pests. Adam optimizer reduces the loss function with a cross-entropy loss to be used for calculating the difference between the adult and true labels. The hyperparameters learning rate (0.0001) and batch size (8) are carefully selected to provide the best training performance in order to avoid the risk of over-fitting the model.

4.1.7 Evaluation Setup

For evaluating the models' results, metrics like accuracy, precision, recall, F1-score, and the AUC (Area Under the Curve) are computed to represent the models' performance. These measures enable one to determine the extent to which the model is able to categorize images of insect pests into a series of classes. The dataset was split into training and testing datasets in an 80-20 stratified split so that the classes were well distributed in both datasets. This is useful in offsetting bias and it is also an advantage such that the model is able to normalize patterns on unseen data.

4.1.8 Working and Communication Tools

To facilitate effective collaboration and communication between the team members, real-time update and communication are carried out based on the tools such as Slack or Microsoft Teams. Also, Jupyter Notebook is utilized extensively to document experiments, visualize results, and for code sharing. This is simple and effective. Iteration and experimentation of a test and train on a model. Jupyter Notebooks are interactive and therefore allow one to iterate and experiment simply and effectively on a test and train a model.

An adequate and flexible framework to train, develop and test deep learning models could be found in the setup of the environment of the present research. This arrangement will make the research productive and effective with the power and strength of its hardware and efficient software frameworks, cloud capabilities, and version control settings to allow flexibility in the future challenge and update responsiveness.

4.2 Testing and Evaluation/Performance/ Comparative Analysis

The investigation has two transformer-based models with the cutting-edge tech applied densely in this work, which are Vision Transformer (ViT) and Data-efficient Image Transformer (DeiT). The models have been found useful in a collection of image classification tasks where images are treated as sequences of patches and hence are able to learn local as well as global relationships in the image.

ViT is a patch model and is composed to simulate association between the patches of an image with self-attention, an concept pivotal in interpreting the background of the images on a global level, a pivotal technique in differentiating pest species which might appear to have some similarity in appearance.

DeiT is also lighter than ViT, trying to construct an improved model from small-sized data via the mechanism of knowledge distillation. Model setup in the paper uses both the models along with the innovation of early fusion

and late fusion approaches. The early fusion technique enables feature maps of both models to be merged at point early in the pipeline, whereas the late fusion technique would add the end predictions (logits) together to make predictions. Prediction bias is also reduced by utilizing majority voting to make the results more resilient. In this work, the IP102 dataset has been used as the source of training as well as the testing of models. This dataset contains images of 102 pest species but the research work is limited to six prominent pest species like *Pieris canidia*, *Potosiabre vitarsis*, Grub, Small brown plant hopper, Yellow cutworm and Yellow rice borer. There are more than 75,000 images; thus, the data is multi-aspect when it comes to showing different conditions, including lighting conditions, resolutions of images, and visibility of the pests. However, because there is a bias when representing the pest species, further data augmentation processes are implemented to ensure the dataset is well-balanced and when it comes to giving equal representation to each of the classes. It includes ground truth annotated images of the pests, which are used for model training.

Preprocessing is a very important process in making sure the quality of the dataset goes into the deep learning models. The resizing of images is done to normalize the images to 224x224 pixels that is consistent with what ViT and DeiT require as the input size of the model. Normalization is done to bring the pixel values of the pictures to be in line with the mean and standard deviation of the ImageNet data to be compatible with the pre-trained model weights. In particular, these pixel values are normalized with the mean of [0.485,0.456,0.406] and a standard deviation of [0.229, 0.224,0.225]. Random oversampling is done to deal with class imbalance, especially on minority classes such that equal representation of all the species of pests is created during training. Random rotation, horizontal flip, resized crops are methods used to augment the data to further increase the data variability, without overfitting the model to the data which makes it generalize well.

The training is performed by relying on the transfer learning of pre-trained models ViT and DeiT and fine-tuning over the pest dataset. Transfer learning also comes in handy to reduce the problem of scarce labeled data since the models will utilize features developed on large-scale data sets such as ImageNet and apply them to the task of pest classification. Training the initial set of the two models is done with pre-trained weights and further trained in pest images. Optimizing is done using Adam optimizer with a learning rate of 0.0001. Early stopping avoids overfitting of the models trained on 50 epochs with batch size 8. The loss function is cross-entropy loss, the best used in cases with multi-class classification where the difference between predicted probability of classes and actual classes are penalized. Training Both early fusion and late fusion are used to combine the output of both ViT and DeiT models. Additionally, the final predictions are subjected to majority voting to generate a better model.

The models' performance will be examined using some convenient metrics for measuring classification performance. Accuracy is the first and main goal for measuring the models' performance. Furthermore, some specific examples such as precision, recall, F1-score will be developed for each pest species so that the model is balanced in terms of false negatives and false positives. Confusion matrix is used to indicate the model's prediction across the various classes and gives the actual and wrong predictions that the model has made regarding the species of pests. AUC is also tested to see whether the model can discriminate among the different classes of pests at different thresholds in such a way that whenever there is any variation in conditions, the model can be applied as it is robust and reliable. Hyperparameters are applied during the training of ViT and DeiT models in a way that the models would be trained in the most appropriate manner. Batch size would be 8 which is a typical choice to train under the deep learning task to get maximum utilization of memory as well as GPU. Learning rate 0.0001, i.e., very less is selected for giving slow convergence, and therefore, does not miss the optimum parameter set. To avoid overfitting, train the models with dropout 0.5 and they are trained for 5 epochs. The model also uses L2 regularization on the weights to ensure that the model does not have gigantic weights and would therefore be at the mercy of overfitting effects. The weights that constitute the late fusion approach are trainingsensitive, they optimize the input of the models at the last decision-making level. These hyperparameters were selected according to deep learning expertise, and they were tuned to be more biased toward the classification of the pest species.

The Early Fusion model also demonstrated superior results in less than 5 epochs and training loss curve revealed that the loss in training was gradually decreasing. Additionally, loss in validation went down in epoch 1:0.271681 and improved to epoch 5: 0.165729. The accuracy of the model was 96.68% and F1 score 0.9667 that affirmed that the combination of features in the early stage was the most effective. It was the Late Fusion model which has been more successful as the losses (training and validation) were sharply decreasing and the accuracy of 97.83% and the F1 score 0.9779 were achieved after 5 epochs. The model used the final predictions of both ViT and DeiT hence it could increase the confidence in the two sources of classifications. The Majority Voting model that had the same prognostics as the other two obtained accuracy of 97.23 percent and F1 score of 0.9723, therefore identifying the strength of the ensemble methods. The ViT Teacher Model that has been pre-trained and fine-tuned on 5 epochs received 98.34accuracy, and the validation loss continued to improve. The Student Model trained using knowledge distillation was also highly effective with a rise of 64.04 to 96.50 percent in its accuracy at epoch one and five respectively.

In figure 4.1 illustrated training vs validation loss of two models Early Fusion and Late Fusion on 5 epochs. The training loss initiates with a reasonably large value of approximately 0.40 and monotonically declines, and at the fifth epoch, its value was about 0.15. The validation loss has the same tendency, initially it was 0.25 and decreased to 0.15 till the very end of the training. It implies that the model is not as evolved yet as it can predict the new data quite well. Nonetheless, the results of the Late Fusion model are a bit different. Here initial training loss was higher and it was approx. 0.25 and it rose to 0.10 at epoch 5. The model commences no less than 0.15 and decreases with no bounce, this implies that the model generalizes perfectly as well. The validation loss paths of two models are similar and so are their training loss paths. At early times it decreases rapidly which is large and at late times the learning is not good (it decreases little), which implies along training and learning.

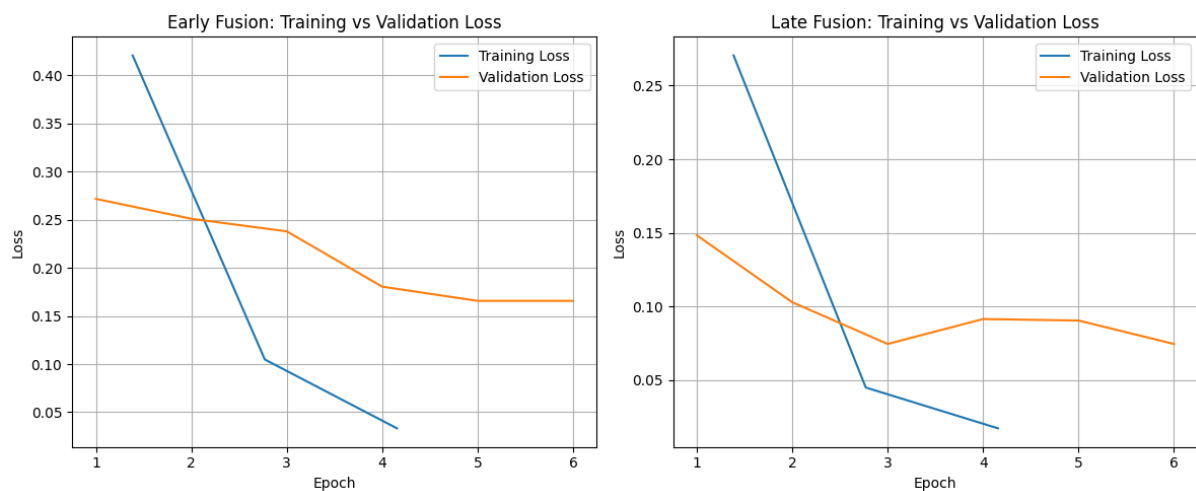


Figure 4.1: Early Fusion and Late Fusion: Training Loss vs Validation Loss

Figure 4.1 represents Training vs Validation Loss of the ViT Teacher Model, EfficientNetB4 Student Model and Loss Comparison of the Teacher and Student Models for 5 epochs. The ViT Teacher Model clearly indicates training and validation loss descent as the training loss value drops from the initial epoch to nearly 0.05 in the fifth epoch and validation loss plateaus to 0.10 which is a great indicator in learning and generalization. However, EfficientNetB4 Student Model illustrates that the reduction in loss follows a declining trend but with training loss increasing to 0.46 to 0.44 while validation loss decreased at a slower rate. Nonetheless, the result does prove that the Teacher vs Student Models Loss Comparison chart illustrates the teacher model leading against the student model more in the validation loss. The student model's validation loss is even greater, yet it is reduced by a great deal over the epoch. The outcome reveals the superiority of the teacher model by exposing its capacity to generalize and learn better than that of the student model that gets a

distillation of the latter but with a lower and middle result in performance.

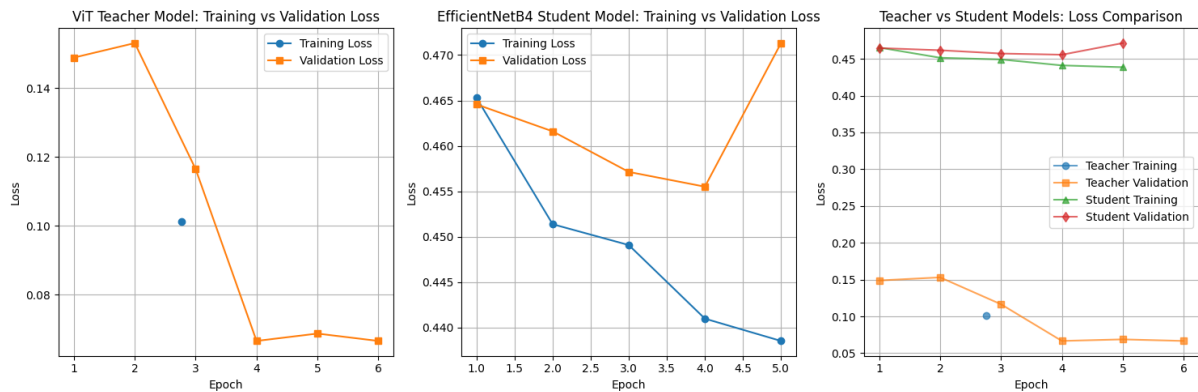


Figure 4.2: ViT Teacher Model and EfficientNetB4 Student Model: Training vs Validation Loss and Teacher vs Student Models: Loss Comparison

The classification report on Early Fusion model demonstrates that 4.1 the model is the elite model at the identification of six insect pest species owing to the very high precision, recall and F1-scores. *Pieris canidia* surfaced as the highest in all the measures (1.0000), hence it is an ideal classification. The precision and recall characteristics of *Potosiabre vitarsis* were 0.9832 and 0.9750 respectively, which provided a very high F1-score of 0.9791 (reliable predictions). The precision was also recorded as 0.9735, recall at 0.9167 and F1- score as 0.9442, which is representative of the good performance that is relatively less consistent. Small brown plant hopper got 0.9741 in accuracy and 0.9339 in recall; hence the F1-score was 0.9536 whilst Yellow cutworm got recall (1.0000) but slightly lower precision of (0.9302); hence the F1-score was(0.9639). Yellow rice borer showed a precision value of 0.9440 and recall value of 0.9752 and F1-score as 0.

Table 4.1: Classification report of Early Fusion

Class	Precision	Recall	F1-Score	Support
Pieris canidia	1.0000	1.0000	1.0000	120
Potosiabre vitarsis	0.9832	0.9750	0.9791	120
Grub	0.9735	0.9167	0.9442	120
Small brown plant hopper	0.9741	0.9339	0.9536	121
Yellow cutworm	0.9302	1.0000	0.9639	120
Yellow rice borer	0.9440	0.9752	0.9593	121
Accuracy			0.9668	722
Macro avg	0.9675	0.9668	0.9667	722
Weighted avg	0.9675	0.9668	0.9667	722

In table 4.2 shows, the late fusion classification report also shows 100 percent of accuracy across all pest species wherein the model gives excellent precision, recall and F1-scores in the plotted diagrams. In the case of *Pieris canidia*, the model had the highest performance in terms of precision, recall as well as F1 score where the value was 1.0000 or error-free classification. *Potosiabre vitarsis* led to the reduction in the recall (0.9833) but the precision with a very high value of (1.0000) and F1 score of 0.9916 and thereby showing the perfect match with the species. Grub exhibited the most performance with precision being 0.9746 and recall being 0.9583 and hence expected F1 score being 0.9664. The recall and precision of the model were high (0.9917 and 0.9524) that gave an F1 score of 0.9717 with a small brown plant hopper. The recall value 0.9917 and precision 0.9835 of yellow cutworm was very high. In the present case, the f1-score was 0.9876. There was good accuracy and recall of yellow rice borer which was 0.9832 and 0.9669 respectively. They introduced a value of F1 of 0.9750.

Overall, the Late Fusion model scored roughly 98.20 % accuracy. The macro and weighted precision, recall and F1-score estimated to be 0.9820 indicate collectively, the overall performance of the model in terms of the success of the model in classifying the insect pests. This implies that the Late Fusion model has the most suitable categorization as it has the greatest balanced accuracy on the different classes and a low misclassification error.

Table 4.2: Classification report of Late Fusion

Class	Precision	Recall	F1-score	Support
<i>Pieris canidia</i>	1.0000	1.0000	1.0000	120
<i>Potosiabre vitarsis</i>	1.0000	0.9833	0.9916	120
grub	0.9746	0.9583	0.9664	120
small brown plant hopper	0.9524	0.9917	0.9717	121
yellow cutworm	0.9835	0.9917	0.9876	120
yellow rice borer	0.9832	0.9669	0.9750	121
Accuracy			0.9820	722
Macro avg	0.9823	0.9820	0.9820	722
Weighted avg	0.9822	0.9820	0.9820	722

In table 4.3 demonstrated the Majority Voting Classification Report indicates how the model performed under which the majority voting ensemble approach was applied. The model had an excellent fit to all the insect pest species. For *Pieris canidia*, it gets ideal out precision (1.0000), its recall is 0.9917 and F1-score is 0.9958 and therefore gives a result which shows the accuracy of this classification to be nearly perfect. *Potosiabre vitarsis* can be said to work in balance because precision and recall both are 0.9833, and F1-score is

0.9833. The grub species performance was accuracy of 0.9573, recall and F1-score of 0.9333 and 0.9451 respectively good performance with low classified errors. The small brown plant hopper had superb recall response of 1.0000, precision of 0.9380 and F1-score of 0.9680, therefore, near perfect identification with only few false positives. The recall and accuracy were excellent with a value of 0.9750 and F1-score also with a value of 0.9750 in Yellow cutworm. Precision, recall, and F1-score was excellent 0.9829, 0.9504, and 0.9664 respectively in the yellow rice borer scenario. Total accuracy of the model was 0.9723 and accuracy it had for all species was indicated by macro as well as weighted average and both were equal to 0.9723.

Table 4.3: Classification report of Majority Voting.

Class	Precision	Recall	F1-Score	Support
Pieris canidia	1.0000	0.9917	0.9958	120
Potosiabre vitarsis	0.9833	0.9833	0.9833	120
Grub	0.9573	0.9333	0.9451	120
Small brown plant hopper	0.9380	1.0000	0.9680	121
Yellow cutworm	0.9750	0.9750	0.9750	120
Yellow rice borer	0.9829	0.9504	0.9664	121
Accuracy			0.9723	722
Macro avg	0.9727	0.9723	0.9723	722
Weighted avg	0.9727	0.9723	0.9723	722

In figure 4.4 demonstrated classification report with the Teacher Model, the scenario in which knowledge distillation is implemented, is succeeded by the implication of high competencies of this model in every category. Apart from this, the model also possessed a good accuracy of 98% which is a sign that the model can identify most of the data upon which it is trained. The recall, F1-scores, and accuracy of all classes are extremely high and all of them average around 0.99, which makes the model excellent at identifying pest species and limiting falsisms and false positives to a minimum. For *Pieris canidia*, *Potosiabre vitarsis*, and yellow rice borer, the model is providing very good accuracy and recall values of 0.99, therefore, verifies that it is capable of identifying these pest species very well and will not be confusing it with any other. Similarly, the small brown plant hopper, grubs, and the yellow cutworms classes pass the category with the model being equally sure of the accuracy and recall values that are all over 0.97. The small brown plant hopper has a recall value of 1.00, which implies that the model has a perfect ability to re-identify this class in situations with all examples with no errors. The final steps of macro average and weighted average also reveal the stability of the model that can be observed due to scores like 0.98 in precision, recall, and F1-score, which can be assumed as just another link in the evidence of the model versatility among classes of pests

Table 4.4: Classification report of Teacher Model.

Class	Precision	Recall	F1-Score	Support
Pieris canidia	0.99	0.99	0.99	120
Potosiabre vitarsis	0.99	0.99	0.99	120
Grub	0.97	0.97	0.97	120
Small brown plant hopper	0.97	1.00	0.98	121
Yellow cutworm	0.98	0.97	0.97	120
Yellow rice borer	0.98	0.98	0.98	121
Accuracy			0.98	722
Macro avg	0.98	0.98	0.98	722
Weighted avg	0.98	0.98	0.98	722

In figure 4.5 demonstrated the general accuracy of the Student Model classification report is excellent at 97 percent. The model was 100 percent accurate (1.00) in two cases, and therefore it attained 100 percent accuracy and a good recall rate (0.99), and this led to an f1-score of 1.00. It enables the model to give correct classifications. Potosiabre vitarsis was identified with highest precision (1.00) and recall rate slightly lower (0.97) which gave it a f1-score of 0.98. Precision of Grub was 0.97 and recall: 0.96, which gives it f1- score 0.97. In the case of Small brown plant hopper, the model was experimented with specificity of 0.93, recall of 0.96 and the f1-score was 0.94. The yellow cutworm model ran with an f1-score of 0.96, a recall of 0.97, and an accuracy of 0.94. That means that the recall rate was quite high compared to the precision rate or in plain words, the model would probably pick up this class. Finally, the Yellow rice borer also had an ideal mark with the f1-score of 0.95, precision of 0.95 and the recall rate of 0.95 having proper balance. Macro average and weighted average both provided the same outcome of 0.97 which showed high concurrence in labeling. Apart from the absence of development foundation, the model may still be estimated with other vexatious species like the Small brown plant hopper on accuracy, recall, and precision.

Table 4.5: Classification report of Teacher Model.

Class	Precision	Recall	F1-Score	Support
Pieris canidia	1.00	0.99	1.00	120
Potosiabre vitarsis	1.00	0.97	0.98	120
Grub	0.97	0.96	0.97	120
Small brown plant hopper	0.93	0.96	0.94	121
Yellow cutworm	0.94	0.97	0.96	120
Yellow rice borer	0.96	0.95	0.95	121
Accuracy			0.97	722
Macro avg	0.97	0.97	0.97	722

Class	Precision	Recall	F1-Score	Support
Weighted avg	0.97	0.97	0.97	722

In figure 4.3 illustrated the early Fusion model confusion matrix is indicating that the model performs highly well in classifying the insect pest species. Precision, recall, and F1-score are all 1.00 with perfect accuracy (120/120) on the model. Potosiabre vitarsis also performs well because it has 117 true positives for correct predictions and 3 false positives for grub cases and thus demonstrates a lesser score in recall and F1-score. For the grub case the model wrongly labels both as Potosiabre vitarsis and small brown plant hopper 2 and 6 respectively so the precision and recall slightly reduces. Small brown plant hopper does contain some misclassifications: 3 as yellow cutworm and 5 as yellow rice borer, otherwise however is very good in general. Yellow cutworm is well classified and yellow rice borer misclassifies a minimal number (3 as grub) resulting in good overall performance. The model, in every class, is highly accurate in classification and fewer misclassifications can be seen especially on the species with the similar visual behaviors.

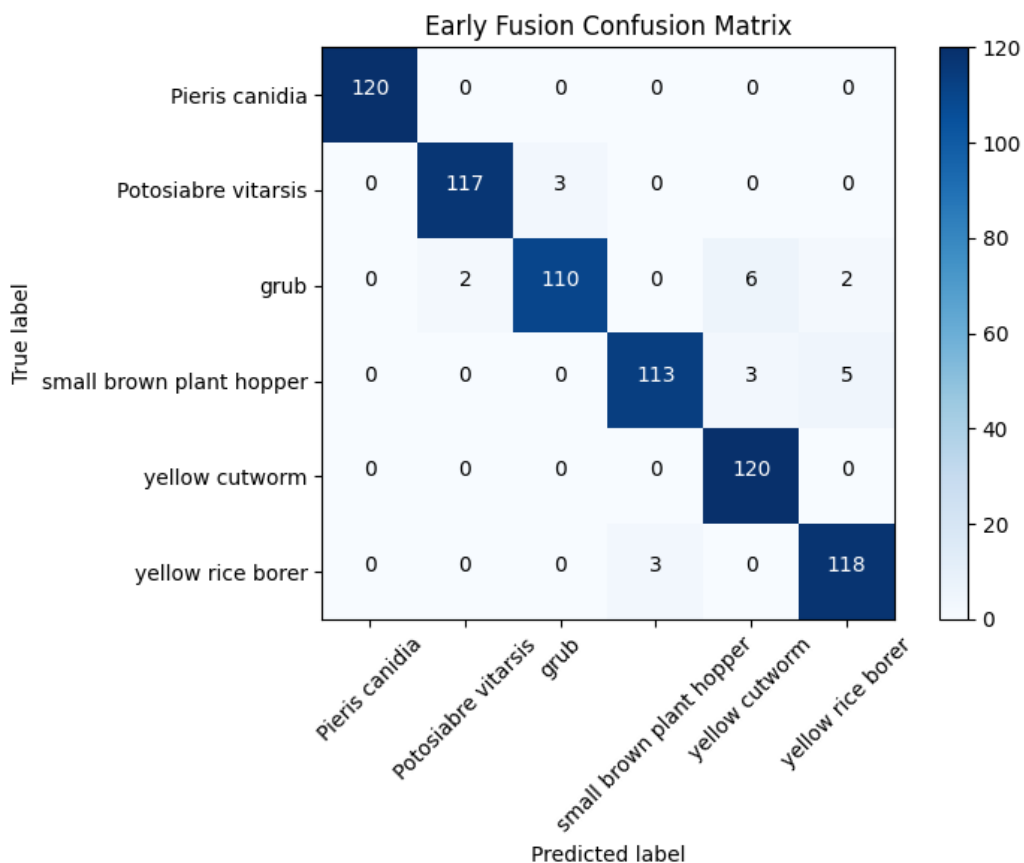


Figure 4.3: Confusion Matrix of Early Fusion

In figure 4.4 illustrated the confusion matrix of Late Fusion is performed well in species classification of insect pests. The model is completely successful in the classification of *Pieris canidia* since there are 120 correct instances classified by the model.

In the same way, *Potosiabre vitarsis* is also predicted with a very high degree of accuracy where 118 predictions have been accurate and merely 2 have been predicted to be grub, reducing its recall and F1-score. Grub observes 115 accurate predictions and 2 classification mistakes: as small brown plant hopper and 2 as yellow rice borer, reducing the precision and recall of the model slightly. Small brown plant hoppers are well classified except for one case which is misplaced as yellow rice borer.

Yellow cutworm is well classified except for one case which was misplaced with grub. The 4 misclassifications which are encountered with yellow rice borer are cases of misclassification with grub, but the model shows a high performance with 117 correct responses. Overall, the Late Fusion model is highly accurate within the range of the categories with little misassignment, particularly on more diverse species of insect pests.

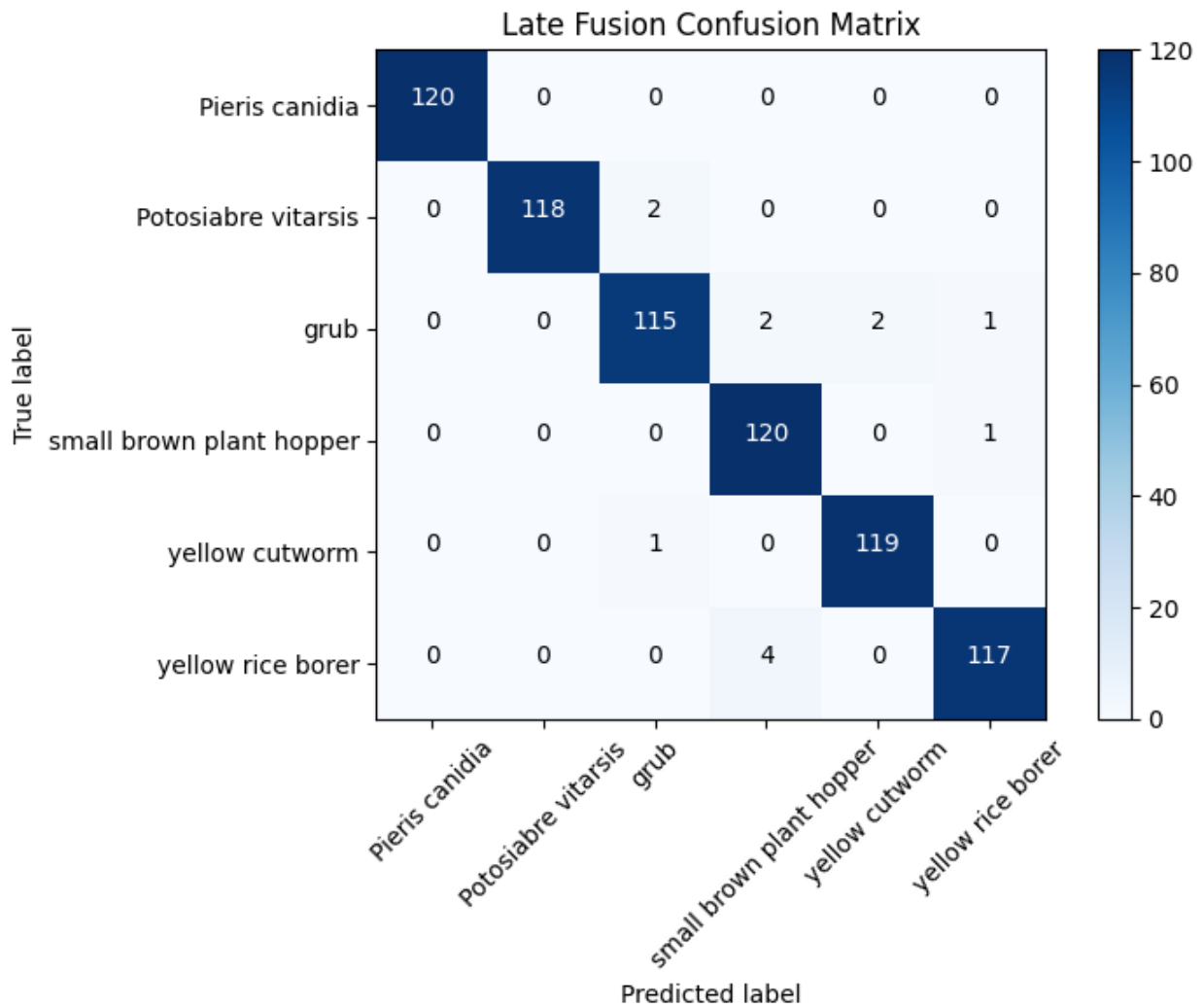


Figure 4.4: Confusion Matrix of Late Fusion

Figure 4.5 illustrates the Majority Voting model class classification in confusion matrix is high classification with misclassification only with a few errors between pest species. *Pieris canidia* is nearly correctly classified (119 correct and with only 1, yellow rice borer), misclassified. *Potosiabre vitarsis* possesses 118 high classifications of which there are 2 misclassifications occurring in grub. In Grub, there is not that high a number of misclassifications and 2 are wrongly classified as small brown plant hoppers and 3 yellow rice borers. Small brown plant hopper is accurately classified in all the 121 complete and without any misclassification. Yellow cutworm has 3 mispredictions as grub, which are not cutting down its result to 117 correct predictions. Lastly, the yellow rice borer illustrates 6 misclassifications on small brown plant hoppers, but the model remains very accurate with a total of 115 correct classification. Generally, the Majority Voting model's performance has been excellent showing little misclassification and capable of distinguishing the majority of the pest species.

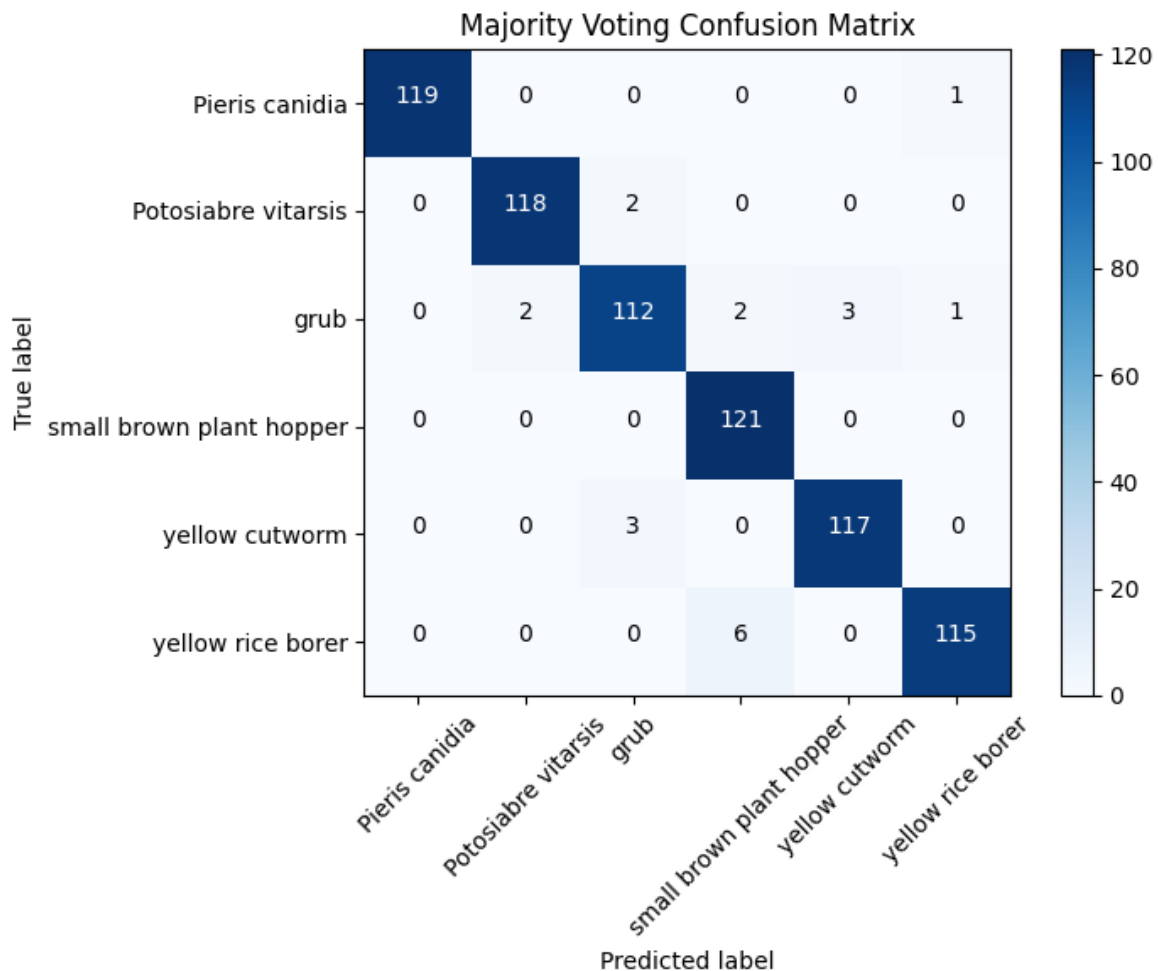


Figure 4.5: Confusion Matrix of Majority Voting Fusion

In figure 4.6 the confusion matrix of Knowledge Distillation Teacher Model will have a very high performance with minimal misclassifications. *Pieris canidia* classification was correct in all but one of the samples that was misclassified as yellow rice borer. *Potosiabre vitarsis* also got a score of 119 correct identification with only 1 misidentification as grub. There was some misidentification under the grub category too, because 1 of the specimens was mistakenly identified as *Pieris canidia*, 1 as small brown plant hopper, 1 as yellow rice borer, but 116 of the specimens were properly identified as grub. Small brown plant hoppers were correctly identified without error, and yellow cutworm was incorrectly identified 2 times (as grub and yellow rice borer) but yet, there were 116 correct hits. Yellow rice borer was incorrectly identified 3 times-both as grub-still, it had 118 hits. In general, the model has achieved a very high performance with minimal mistakes and this, especially with the classification of most of the species demonstrates the strength of the teacher model in knowledge distillation tasks.

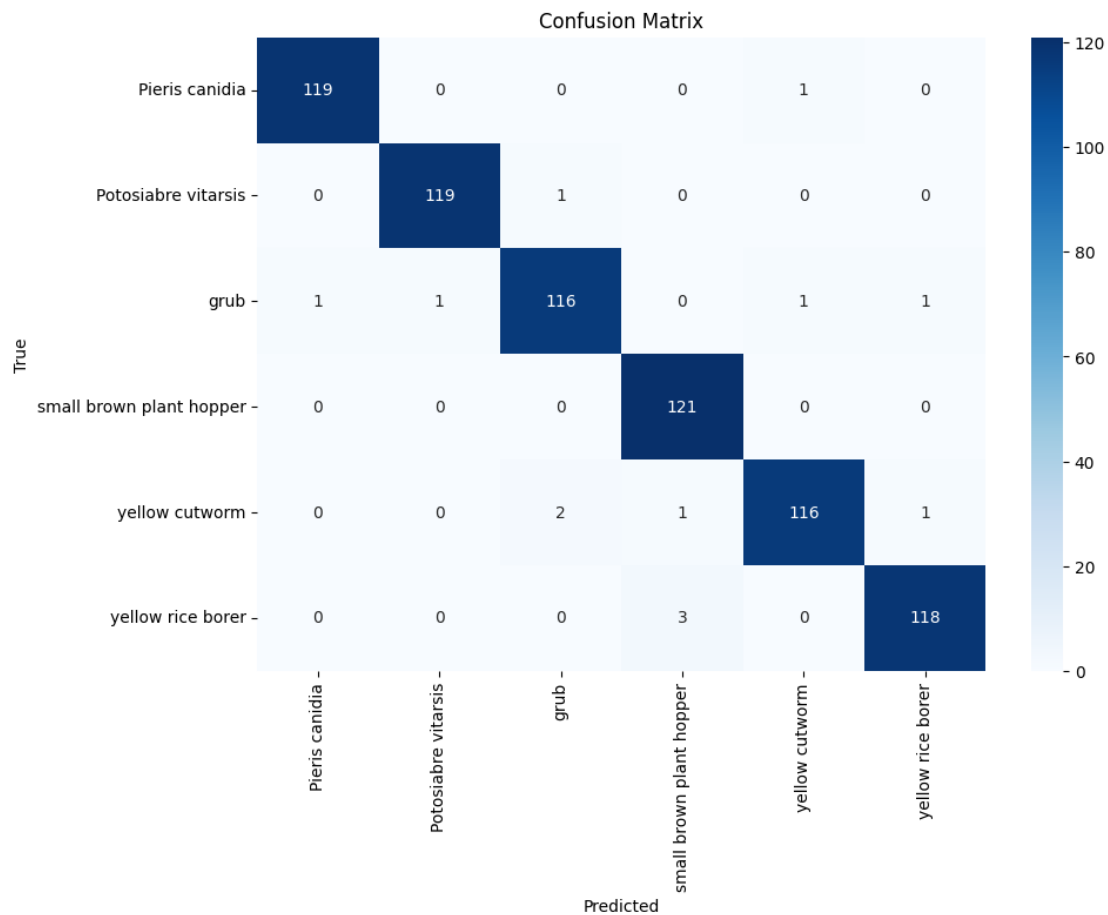


Figure 4.6: Confusion Matrix of Teacher Model (KD).

Figure 4.7 illustrates the confusion matrix of the Knowledge Distillation Student Model indicates that the model can achieve a good accuracy percentage in most of the classes or categories but there were some errors. In nineteen out of 119 cases, *Pieris canidia* was identified correctly and in one case it was identified as yellow rice borer incorrectly. *Potosiabre* identified correctly 116 times, and mislabeled 2 times as grub and 2 as yellow rice borer. In grub, it identified correctly 115 times but mislabeled 2 as small brown plant hopper and 3 as yellow rice borer. Small brown plant hoppers were identified correctly 116 times, 2 were confused with yellow cutworm and 3 with yellow rice borer. Yellow cutworm had 1 misclassification as grub and 2 misclassification as yellow rice borer and yet, 117 were correctly classified. There were 6 misclassifications in yellow rice borer with the model correctly predicting 115 grub instances that were predicted in this class. In general, the student model is doing very well considering the minimal errors, an indicator of the fact that the knowledge distillation process between the teacher model has been successful in passing useful features and therefore achieving correct classification, even in the majority of the pest

species.

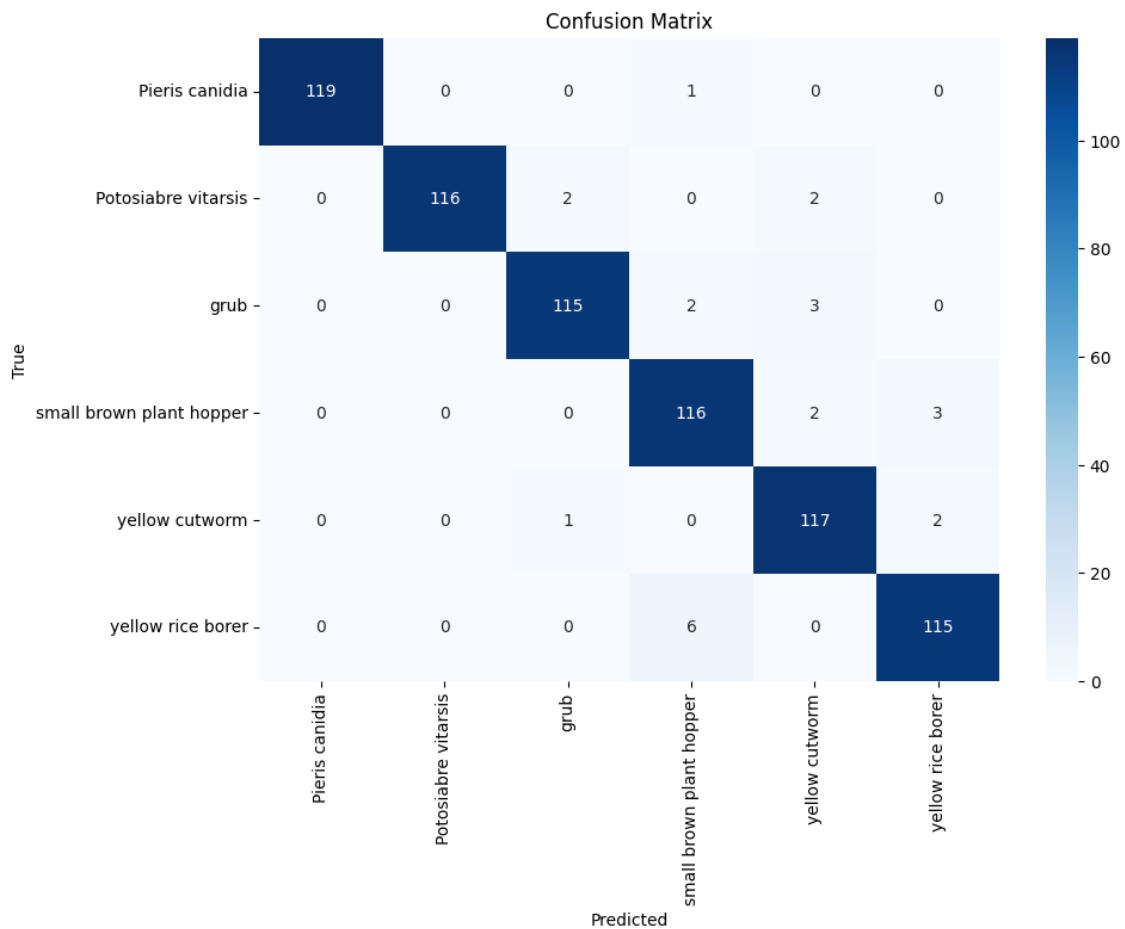


Figure 4.7: Confusion Matrix of Student Model (KD)

Figure 4.8 illustrates the ROC curve comparison graph shows how stellar performance of the three models Early Fusion, Late Fusion, and Majority Voting with a performance of 1.0000 on the AUC scale has an error-free classification. The TPR rises dramatically when the FPR is negligible, indicating that such models can correctly classify the species because the pest species are classified correctly. To provide a baseline and to indicate the random classification accuracy that is much poorer than for the fusion and majority voting models' usage is made of the Chance line, represented as a dashed blue diagonal. The manner above the opportunity line of the voting and fusion models' curves indicates further evidence that they provide higher performance than others in order to distinguish between the pest classes. In general, one can see that all three models have ideal performance, and the classification of the pests is reliable and provides accurate classification in comparison to chance guessing.

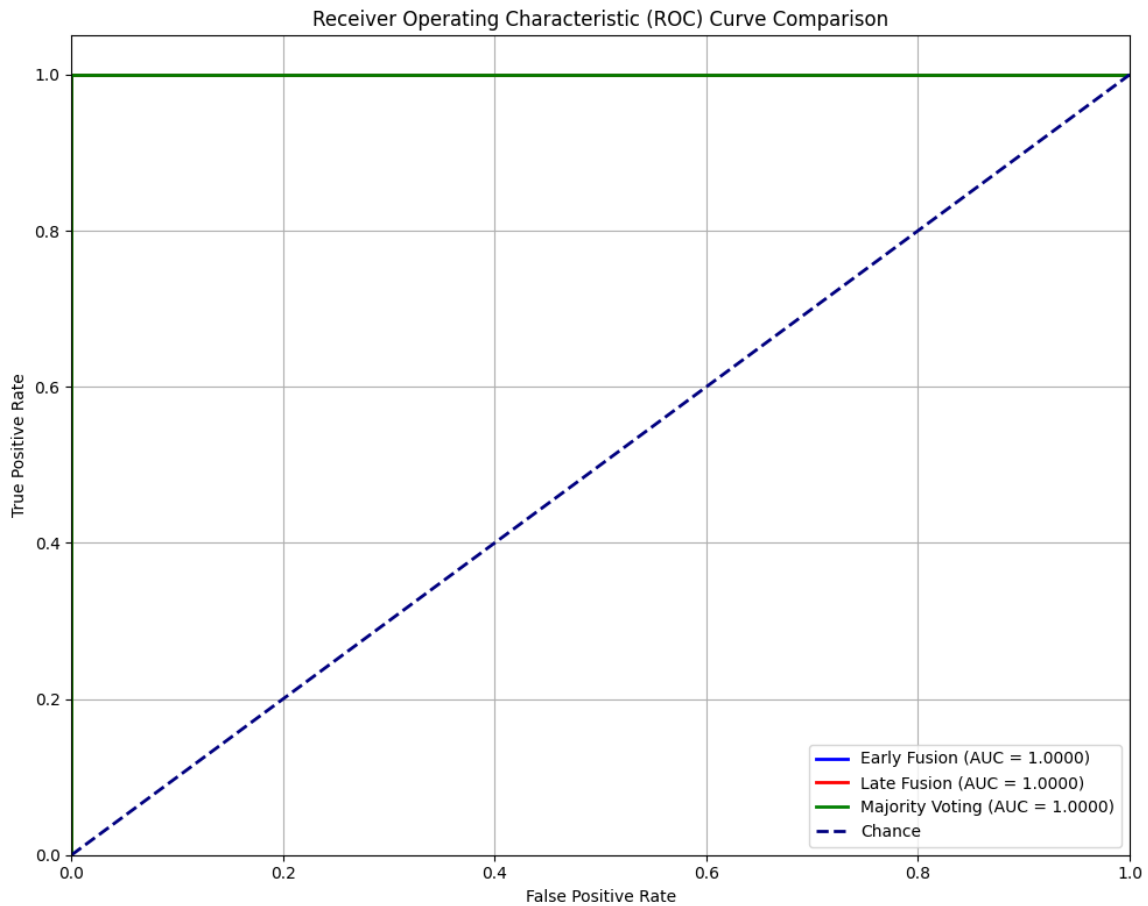


Figure 4.8: ROC Curve of Early Fusion, Late Fusion, Majority Voting, and Chance

In figure 4.9 illustrated the ROC plot presented here is for the Knowledge Distillation Teacher Model, the result is AUC (Area Under the Curve) of 0.99. The curve shows the proportion of True Positive Rate and the false positive Rate. Steep upward trend towards the top-left means the model is efficiently separating the various classes with a high True positive and a low False positive rate. The model nearly gets the best classification because the curve is at the top-left corner which indicates there is a perfect classification. The Chance Line with dashes appearing in blue is a plot of the baseline performance of an arbitrary classifier upon which the teacher model performs very well. Through the AUC, the value of 0.99 also shows the well workability of the model as it is a good tool of identifying the target categories with a very high degree of precision.

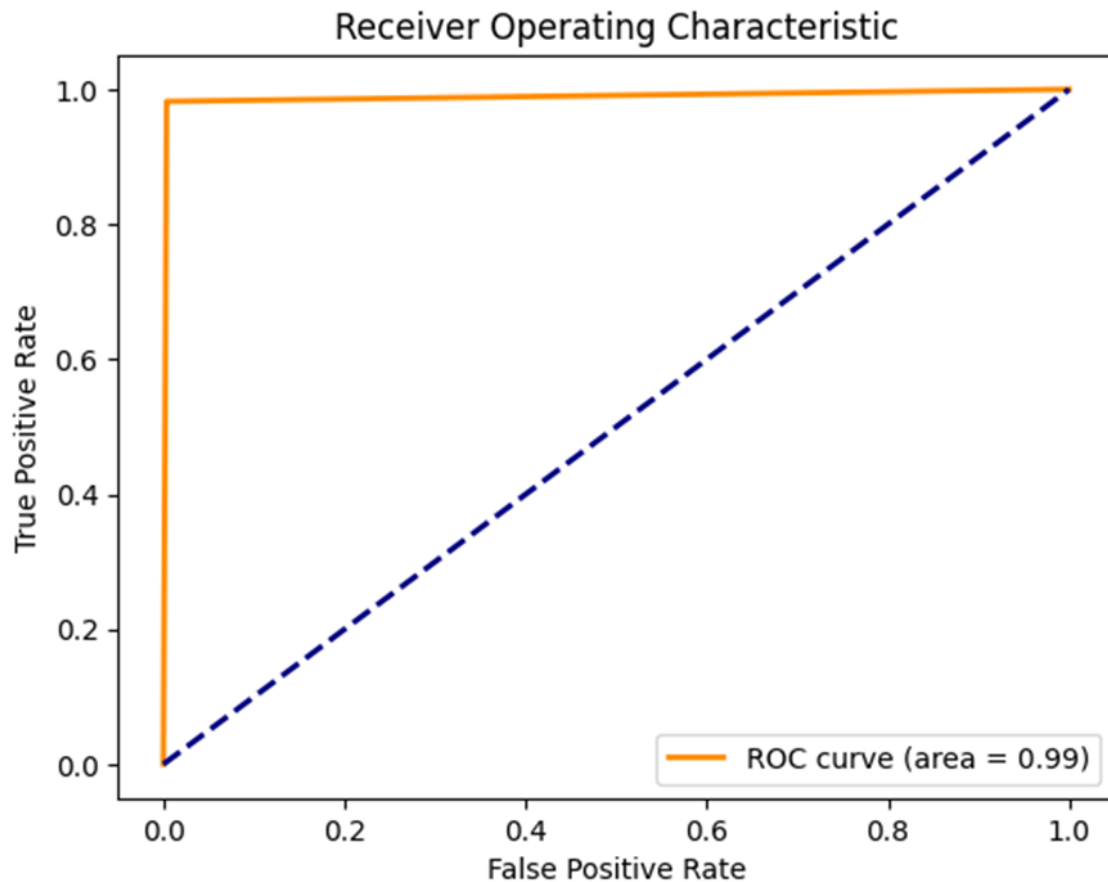


Figure 4.9: ROC Curve of Teacher Model (KD)

In figure 4.10 illustrated the ROC curve of the model is represented as the Knowledge Distillation Student Model, which has a performance indicator (AUC) (Area Under the Curve) of 0.98. Like the teacher model, the performance in this student model is also doing very well, meaning that this student can easily differentiate classes as shown in the rapid incline at the top left corner of the curve. The curve indicates a high True Positive Rate (TPR) which grows substantially with a False Positive Rate (FPR) that is nearly zero and the graph is high and the classification efficiency is high. The fact that the AUC score is 0.98 is evidence that the student model is performing outstandingly, but not as perfectly as the teacher model as its AUC score is 0.99. The dashed blue line is the Chance Line and it indicates the random guessing baseline and the student model curve rises well above that. This implies that this smaller and more efficient model of student remains efficient since the transferred knowledge of the teacher model is still capable of yielding good performance.

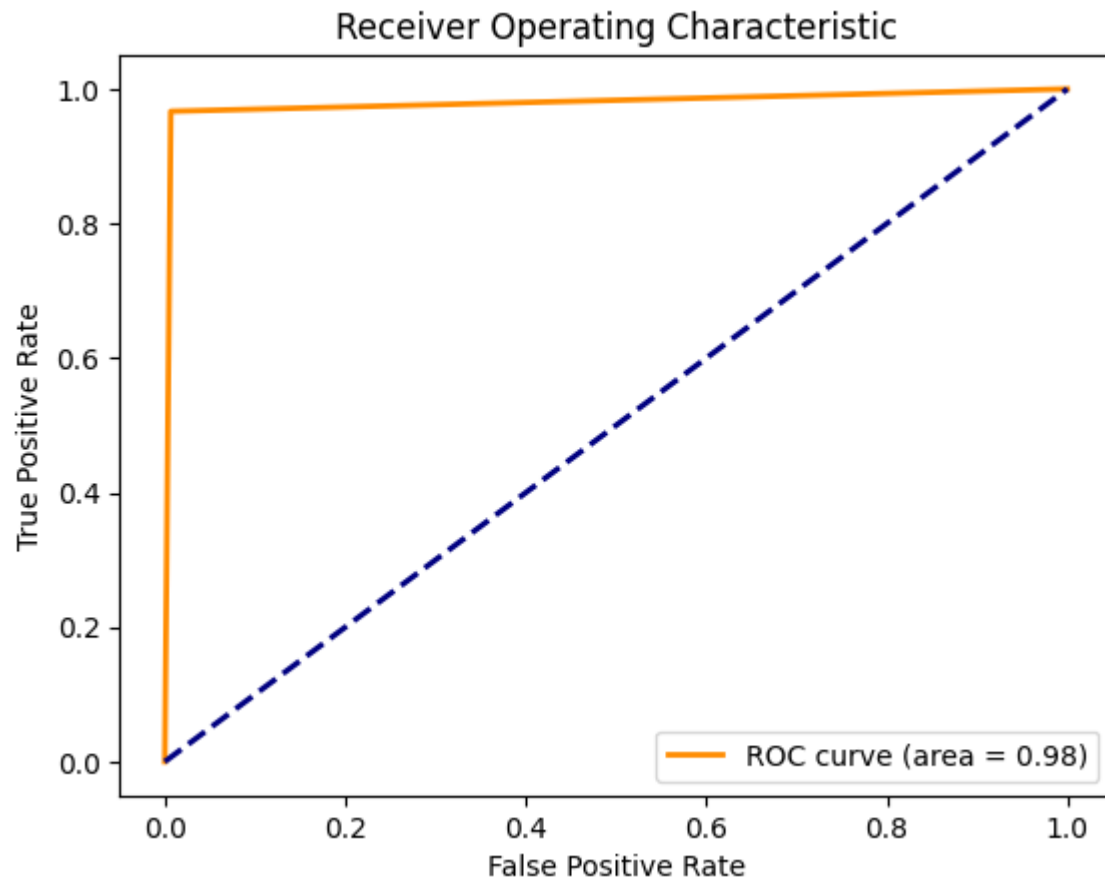


Figure 4.10: ROC Curve of Student Model(KD)

Figure 4.11 illustrates the preceding bar chart contrasts the accuracies of the different models, i.e., Early Fusion, Late Fusion, Majority Voting, Teacher Model (KD), and Student Model (KD). Knowledge distillation, i.e., by the Teacher Model (KD), yielded the maximum level of accuracy 98.33%, thus reflecting the overall advantages of using data with a larger pre-trained network to surpass its peers. Late Fusion scored 98.20% accuracy, higher than other ensemble techniques, an indication that it performs exceptionally in combining predictions of various models. Majority Voting performed quite well with the performance being 97.23 percent while in Early Fusion performance was not good with 96.68% being exactly the worst among the fusion methods. The trained KD was the Student Model with a 96.50 % accuracy, which was not good in comparison to the Teacher Model but outstanding, especially in terms of its computational efficiency. In general, Teacher Model and Late Fusion worked best and Student Model had fair possibilities of deployment in a resource-constrained environment.

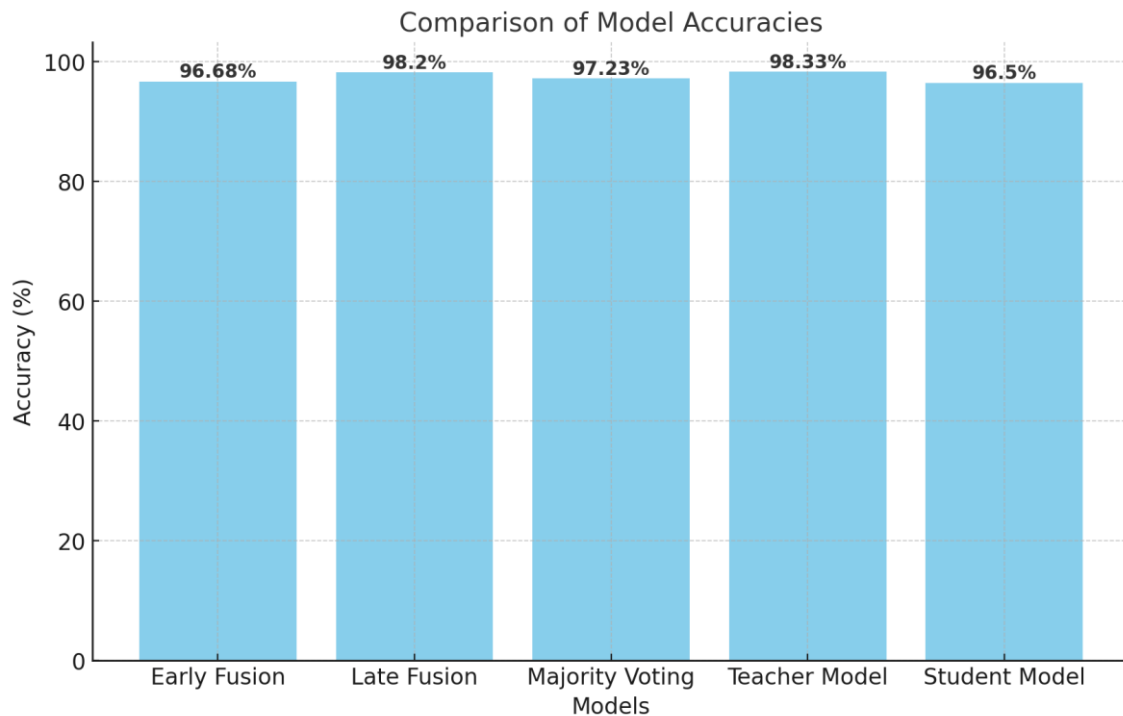


Figure 4.11: Comparison of accuracy between all models.

In figure 4.12 illustrates the comparison of the difference of value of the loss of various research models. The negative value of Early Fusion is 0.1657 which is not too big but still spread down into the rest of the models. However, the value of loss 0.0901 with Late Fusion is smaller and may be regarded as efficient in comparison to the reduction of training loss. More optimistic results can be noticed in the Teacher Model with the value of loss 0.0688 reflecting that the model is extremely proficient to solve the problem of classification. However, the Student Model on knowledge distillation training depicts a significantly large value of loss with the value of 0.4705. This would imply that each time there is any existing realized merit of distillation, the Student Model is faced with the challenge of optimization that results in the pursuit of the increased loss in comparison to its Teacher counterpart. Teacher and Late Fusion average better performance in loss and Student Model, which however does not run as many computations as towards the same level of performance.

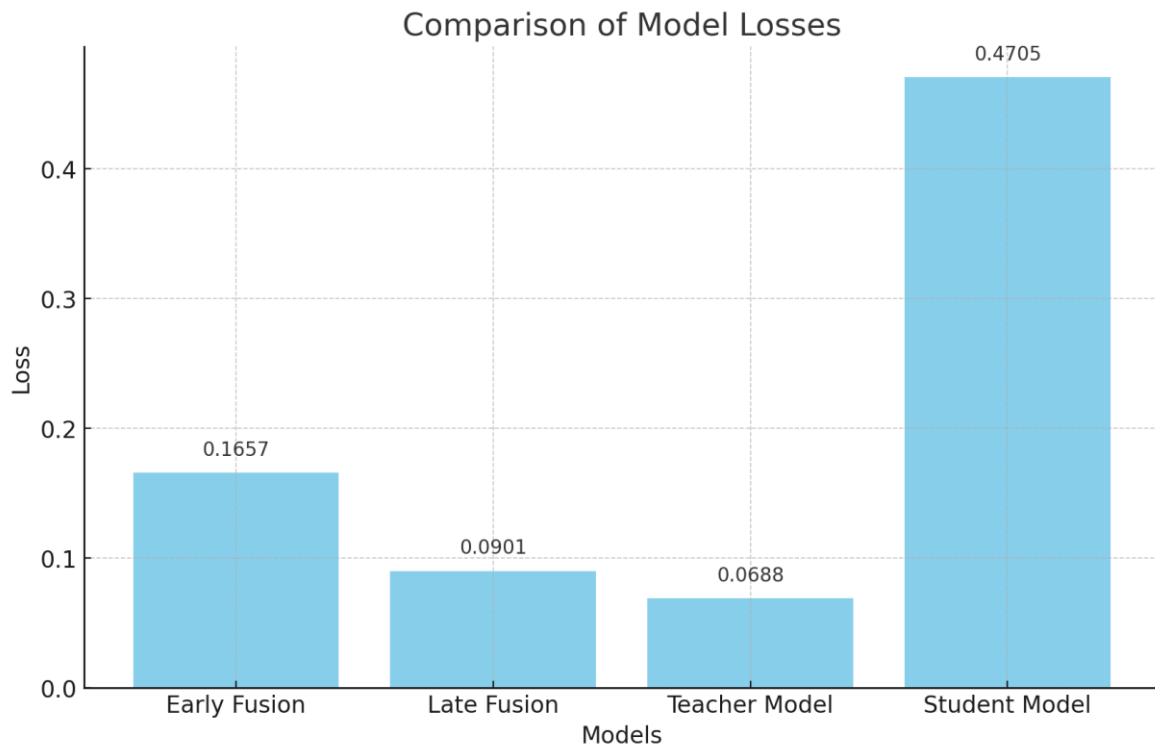


Figure 4.12: Comparison of loss between all models.

Early Fusion, Late Fusion, Majority Voting, Teacher Model (KD), and Student Model (KD) models were compared to identify notable differences in view of different performance measures, i.e., accuracy, loss, precision, recall, F1 score, confusion matrix, and ROC curve.

Late Fusion was equally the winner by its accuracy of 98.20 followed by the Teacher Model with 98.33. Yet, in loss, the Teacher Model was significantly more efficient with loss less (0.0688) than Late Fusion (0.0901) or Early Fusion (0.1657) and significantly higher than the Student Model at 0.4705. This means that the Teacher Model worked better in reducing the errors in numbers, while the Student Model had more errors, especially in dealing with complex classification of classes. Both the Teacher Model and Late Fusion performed extremely well for precision, recall, and F1 score for all the pest categories where Late Fusion had slightly high tapering precision and recall balance which ultimately gave it a good F1 score overall. Although the Student Model (KD) generalized very well, there was evidence of lower precision in some of the categories because it was not able to predict the answer of some classes specifically yellow rice borer as frequently as it was found in other classes which implied that there was not good generalization of the Student Model (KD) to less frequent classes. Teacher Model and Late Fusion had very low misclassification particularly on such as

Pieris canidia and yellow cutworm thereby being able to differentiate accordingly in terms of classes. Student Model, however, misclassified on repeat, particularly on yellow rice borer, but still to a very acceptable extent on other groups. The ROC curve analysis supports the perfect performance of the Late Fusion and Majority Voting with AUC score of 1.000, which means that they perfectly classify. The Teacher Model recorded a very close result just 0.99 as compared to the AUC of 0.99 by the Student Model that was little behind suggesting that it is comparatively more inadequate to be able to classify some pest species correctly.

Late Fusion was the most powerful and gave the highest accuracy and better balancing among metrics and the Teacher Model had a high level of precision and low loss, hence a very efficient model. Majority Voting performed particularly well in accuracy too though the Student Model was a bit worse off especially in the manner it dealt with difficult classes due to effects of knowledge distillation.

4.3 Results and Discussion

As earlier research studies have revealed, the highest average performance was that of the Late Fusion model whose accuracy rate was 98.20%, closely followed by the Teacher Model (KD) at 98.33% accuracy. The two models were nearly at par in terms of their accuracy, recall, and F1-score, with the best composite of the three being the Late Fusion model. Therefore, the most stable of the three measures was the Late Fusion model and thus the best with regard to accurate identification of the pests. The Teacher Model (KD) with high loss performed better especially in precision and recall classification, that is the model was able to distinguish between the pest species with minimal bias of misidentification. Being of lower computation cost, Student Model (KD) was extremely accurate at 96.50 percent but not effective in detecting pest species, at least in terms of being less prevalent (more Yellow Rice Borer) because of the very inherent nature of knowledge distillation. The Majority Voting model, however, performed quite well, as evident through the performance of both the DeiT and ViT output fusion as the model had an accuracy of 97.23%. The early Fusion model, with 96.68% accuracy, performed the worst; i.e., early-level fusion did not bring a massive improvement in comparison to other models. Aside from these results, the ROC curve results supported these results since the Late Fusion and Teacher Model (KD) achieved a perfect classification with an AUC of 1.000 and the Student Model (KD) was 0.98. The results support that ensemble approaches, Late Fusion and Majority Voting, are effective in improving the performance of classification, and that Teacher-Student Knowledge Distillation is effective in precision vs. computational speed trade-off. These findings have the potential to contribute positively towards efficiency of different fusion techniques and model optimization techniques, particularly pest classification and corresponding relevant diagnostic issues in low-resource environments.

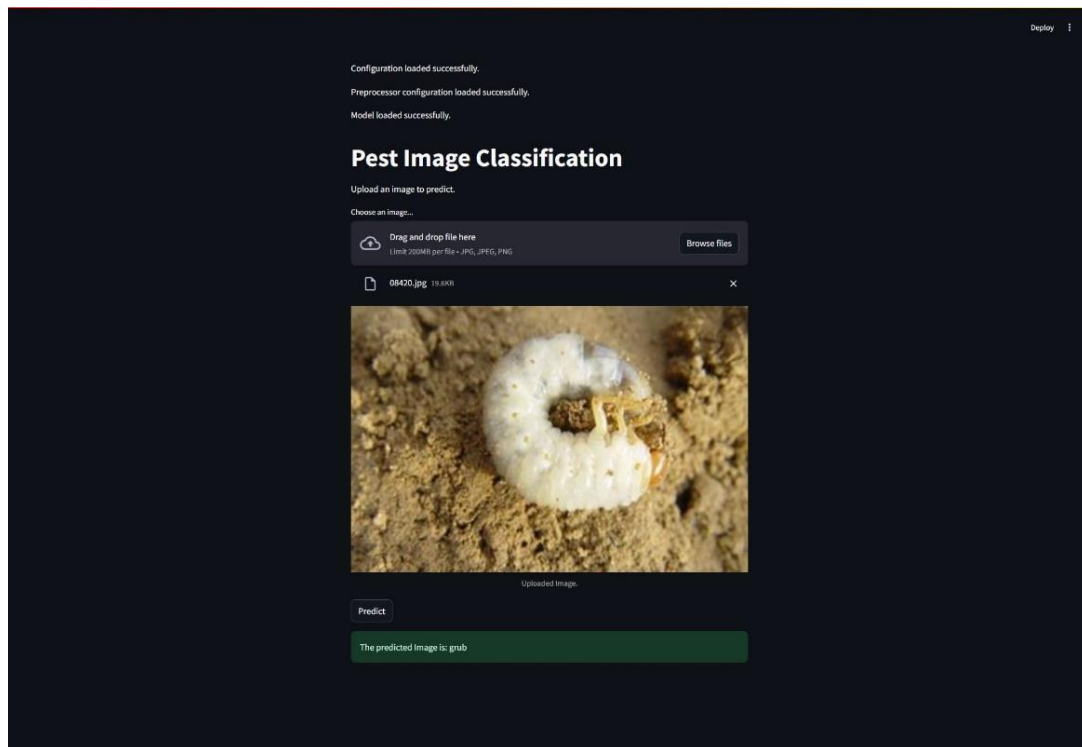


Figure 4.13: Web Application.

4.4 Summary

The research infrastructure will comprise a high-end hardware with an NVIDIA RTX 3080 Ti graphics card and a minimum of 100 GB of storage capacity of the dataset and the model checkpoints. The software environment interprets Python and deep-learning frameworks such as TensorFlow, Keras, and PyTorch, as well as data-handling and visualization libraries such as NumPy, Pandas, and Matplotlib. ImageDataGenerator and albumentations are then used in preprocessing the IP102 dataset with resizing, normalization, and data augmentation. Git and GitHub are used to control version and collaborate. ViT and DeiT models are pre-trained and then fine-tuned on the IP102 dataset with the Adam optimizer, and a learning rate of 0.0001. Measures of performance are precision, recall, accuracy, F1-score and AUC. This arrangement provides effective training, evaluation and mass implementation of the insect pest detection system.

Chapter 5

Engineering Standards and Design Challenges

This chapter talks about the standards observed by engineering in the process of engineering the pest detection system including the hardware, software and communication standards. It also relates the issues of design encountered when working on the project such as accuracy-efficiency trade-offs and class-imbalance and real-time performance to be applicable to deploy in resource-limited settings.

5.1 Compliance with the Standards

Best practices and optimal techniques in deep learning, computer vision and model optimization were followed in carrying the project. The data that was used was subjected to data preprocessing to make it usable by the ViT and DeiT approaches; this was done considering resizing, normalization, and data augmentation. The testing of these models is done on traditional metrics such as accuracy, precision, recall, F1-score and AUC among which strict testing is underway. Ensemble learning and improving the robustness of the model by enhancing the fusion methods, viz., early fusion, late fusion, and majority voting is according to the norm of the industry. The compact efficiency model among the EfficientNetB4 model is fine-tuned with knowledge distillation technique where the computational power is consumed at minimal levels and the optimization level of precision is established. The ethics of data usage is upheld and the models are efficient and deployable in real-time and have minimum utilization of resources particularly for mobile or edge devices. The efficiency problem of computing, the class imbalance, the overfitting problem and the data insufficiency issues are addressed in the advanced techniques of regularization, transfer learning, and data augmentation. Overall, the proposed method will provide the cutting-edge deep learning methods and the industry standard widely used ones so that they may combine their efforts and deliver effective pest identification within the limited environment.

5.1.1 Software Standards

The software standard deployed was the standard used in this research that has been recommended in the industry about how machine and deep learning models should be migrated. Python has been the most notable programming language, being selected because of extensive libraries and frameworks, which assist in the possibility of carrying out machine learning tasks. Examples of libraries that are being utilized in the definition and training of models include TensorFlow, Keras, and PyTorch since they are well

established and sponsored and will be scalable and resilient in any environment. Pandas and NumPy libraries are employed to perform data operations and manipulation which include efficient data structures and operations when it is necessary to process very large datasets. Seaborn and Matplotlib are used to create plots that have meaning when the task at hand is data visualization to understand performance. The model training involves a GPU-compatible environment in which it utilizes CUDA and cuDNN in order to execute expedient train calculations. Git and GitHub are introduced to provide the version control and collaboration capabilities so that they can provide a methodical means to document the transformations to codes, make them reproducible or maintain the work of the other team members. The fact that such software standards are observed will ensure that there is system reliability, maintainability, and scalability, the ease of teamwork, and the integration of this system into other systems and tools.

5.1.2 Hardware Standards

The requirements of the devices and the hardware of this work can handle the high-performance requirement that is necessitated in the deployment of the training and deep learning design. To bring out a quicker computing time on big datasets and intricate neural network layout like Vision Transformer (ViT) and Data-efficient Image Transformer (DeiT), they comprise High-end Graphics Processing Unit (GPU) e.g., NVIDIA RTX 3080 Ti in training the models. The parallelism on capability of the GPU is very high and it would adversely affect the time-efficiency of training since it can proceed with the computations of many images and model iterations also. It wastes too much storage space in storing the IP102 dataset, model checkpoints and other intermediary data to carry out the training and evaluation process more than 100G inclusive which does not occur. This confirms the registered and secure place of storing the dataset and model parameters that can be accessed with ease in future during the processing and analysis. The RAM used to install the system is adequate (usually 32GB and above) such that it has boundaries on the amount of memory it can handle regarding the generation of data and training. Such hardware requirements will show the perfect performance, scalability, and reliability of the deep learning models involved in the classification of the pests present in the researching environment.

5.1.3 Communication Standards

Hence, good communication is among the topmost things upon which an inquiry can be carried out effectively even with the complicated models and data sets available. Under research context, there are communication norms that are set to assist in organizing the collaboration within the company and govern understanding of all the participants by newly emerging information regarding progress being made. Its primary means of communication is a messaging application in real-time such as Slack, Team Microsoft

that facilitates fast and efficient communication among different members of the team regardless of their location. The success of such arrangements is the ease of text message, voice and video conferencing, which ensures that technical meetings, briefings and troubleshooting continue uninterrupted. Moreover, version control through the support of GitHub is also facilitated to ensure that the employees could trace the modifications made to the codebase, exchange updates, and finish the coding assignment. This is standard procedure that ensures that every member of the team receives the updated code and that there ought never to be any inconsistency between those versions. To capture and display the data, Jupyter Notebooks are utilized, by means of which one can create interactive documents wherein the code, visualizations and text description are added. It would ensure experimental results will be properly documented, and it will be easy to share with the team.

In order to further ease the collaboration, various meetings and brainstorming sessions are held, progress is discussed, obstacles are sorted out, and follow-up appointments are made. With such a communication system, the research team had an opportunity to arrange the amount of operations needed, timely notifications, and the astounding level of transparency about all the following phases on the project.

5.2 Impact on Society, Environment and Sustainability

The proposed study and the subsequent pest detection system can produce effects back to society, the environment and sustainability through a number of ways. This study indicates how a deep learning-based system (insect pest detection) can combat an important agricultural problem, i.e., effective and timely detection of the pests threatening crops. The pest can be detected early hence farmers act promptly by putting the right measures in place to curb the flea of pests hence resulting in minimal use of pesticides. Not only does this help the environment in a positive way (reducing the discharge of chemicals and eroded soil) but also has the positive effects of contributing to improved food safety and quality, ensuring that little to no pesticides are over-utilized in agricultural processes. At a societal level, novel systems are useful especially in constrained environments with resources where pests may be detected traditionally. Small-scale farmers and communities in the rural area, through the mobile-friendly technology can gain access to cutting-edge tools to manage pests by implementing an affordable, efficient and effective system to detect them. This not only assists in enhancing the food security of a country through increased harvest and crop harvest loss but is also essential in the life of farmers and in the larger economy of any economy which involves agriculture.

Regarding sustainability, the study postulates the process of precision agriculture, in which the resources are utilized better, and an environmental imprint is minimized.

Knowledge distillation and fusion coupled with the implementation of deep learning models such as ViT and DeiT means that the system can be computationally efficient and run in environments where there is only limited availability to high-end computing systems. Besides lowering the dependency on using chemical pest control practices, the system contributes to the sustainability of agricultural products, which can lead to retention of biodiversity, pollution reduction, and ecological stability in the long term. On the whole, the study can enhance agricultural output, promote economic growth, and even environmental protection and are in line with the sustainability objectives worldwide.

5.2.1 Impact on Life

The artificial intelligence-powered pest detection system that is proposed can make a significant difference in the quality of life in different industries and in particular, in the spheres of agriculture/food security and the health care system. The system allows farmers to take immediate actions and thereby prevents large-scale crop destruction by offering a credible and time-saving means of early identification of an insect pest. This has the potential to raise agricultural production, improve agricultural products quality and incomes among farmers especially in the developing world where the economy thrives on agricultural activities. The greatest beneficiaries of a mobile-friendly tool capable of identifying pests at a considerably low cost would be small-scale farmers who may not necessarily have the means of accessing sophisticated ways of controlling pest infestation. Also, the system aids in the maintenance of the health of people because it helps to avoid pests on crops as they may be disease-transmitting pests like diseases that affect the human health of animals. Early detection of pests minimizes the use of the dangerous chemicals as pesticides making the food products safer both in terms of health and exposure to some toxic elements as to the farmworkers and consumers. Also, it will have positive effects in the environment since the system contributes to more sustainable methodologies in the agricultural industry. The pest detection system contributes by reducing the use of pesticides, conserves biodiversity, helps to maintain natural habitats and soil and water pollution which were the by-products of using excessive pesticides. In a nutshell, this system has a positive impact on the lives of farmers, consumers, and the community and enhances food security, health, and sustainability of the environment. The achievement of its implementation can translate to a stronger agricultural sector and healthier ecosystem, which eventually makes people and the planet healthier.

5.2.2 Impact on Society & Environment

The projected deep learning-based pest detection system will impact society and the environment with far-reaching positive implications. Socially, the system can revolutionize the agricultural business sector with the realization of a trustworthy, high-

achieving, and inexpensive method of managing pest infestation by farmers. Since the pests are simpler to detect, farmers will experience fewer setbacks, minimize the negative impacts on their crops, and ultimately enhance food supply that is an essential need to feed all people as the world population increases. This would also enhance the livelihood of the farmer, especially that of the resource poor set by providing them with an affordable method of managing pests without depending a lot on costly chemical insecticides. An opportunity that the system possesses also involves addressing public health issues by mitigating the risk of use of toxic chemicals in agriculture. Pesticides can have adverse effects on foods, water bodies, and the environment, either to consumers or to laborers and people around the area. The system reduces the risks by promoting the application of environmentally sound methods of pest management in a bid to promote safer food production and a healthier society. Additionally, it maintains useful organisms as well as pollinators, which are vital for ecosystem fitness and biodiversity.

Environmentally, the pest detection system is eco-friendly in that it minimizes the application of chemical pesticides that have been established to have adverse effects on the welfare of soil, contaminate water bodies and cause harm to wild animals. The application of the technology will lead to greener agriculture where agricultural activities are not at variance with ecosystems' sustainability in the long run. Apart from this, the system helps in preventing overuse of chemicals, thereby limiting potential harm to the environment since they can be applied with utmost accuracy. Therefore, the above system is not only an added factor towards agricultural productivity but leads to conservation of the environment and well-being of contemporary society. Its capacity to identify pests early and correctly presents it as sustainable technology, in view of its benefit to the society and the environment thus a major contributory factor in the sustainable development process of agriculture and far healthier world.

5.2.3 Ethical Aspects

The proposed pest detection system based on deep learning has a few ethical concerns that would have to be considered to use it responsibly. To begin with, fairness and accessibility is of prime importance since the technology has to be affordable and accessible to not only the large-scale farms but also the small-scale farmers in the resource limited regions. The system may increase the given disadvantages without equal access. It should also be open and accountable since in most cases learning deep models are regarded as black boxes, and it is crucial that farmers know how decisions are determined, particularly when the model fails. Another issue is data privacy because the imagery data of the farms have to be anonymous, and the data needs to be treated as the personal data of the farmers. Besides, the environmental impact of the system should be included, especially the amount of energy used when training a deep learning model and deploying it. There should be endeavors to see to it that the technology

encourages sustainability without causing unnecessary environmental degradation. Lastly, the phenomenon of bias within the data utilized by the model will have to be tackled, as biased data would result in a wrong identification of the pest, and subsequently, damage to a particular region or crop. Taking into consideration these ethical dimensions, the pest detection system can be created in a manner, which can be of advantage to the farmers and the society, as well as encourage equity, disclosure, and ecological accountability.

5.2.4 Sustainability Plan

A long-term social and environmental sustainability-based sustainability plan for the proposed deep learning pest detection system exists. For environmental sustainability, energy can be saved by making the system suitable for low-resource areas. This is achievable by employing energy-efficient models like EfficientNetB4 as the student model, as opposed to larger resource-based models like ViT. Besides that, cloud platforms are also utilized to train and refresh models to minimize the workload of users' hardware in low-resource environments. For economic viability, the system should be made affordable and accessible to smallholder farmers and preferably subsidized or in collaboration with agricultural institutions to render its utilization less expensive. The models need to be developed deployable using low-cost devices such as smartphones or edge computers to ensure that they are deployable to rural areas, where farmers will have no easy access to high-performance devices. For social sustainability, the system should be community-based in nature and provide training and assistance to the people of the community in such a way that they can utilize the technology up to its maximum level. This can be achieved by designing easy interfaces, providing assistance in the local language, and providing usage guides of the technology so that the usage of improved pest control measures, and hence the crop yield, is enhanced. Finally, the system needs learning feedback. Frequent update from users and new information on pests will keep the system updated and relevant. The AI and machine learning models need to be responsive to changes in agricultural ecosystems and adaptive techniques of pests so that the system continues serving the agricultural society through and on to solving new problems. With ecologic, economic, and social sustainability, this pest detection system can potentially be an advantage for farmers globally to utilize in sustainable agriculture with lasting effect.

5.3 Project Management and Financial Analysis

Provide a cost analysis in terms of budget required and revenue model. In case of budget, you must show an alternate budget and rationales. The deep learning based pest detecting system would be only successfully built under the management of effective project management. The project is planned to have several phases with

its set of tasks, time frame, and resources. These stages are research and planning, collection and preprocess of data, model development, testing and evaluation, deployment and integration as well as documentation in the final report. A project manager will also coordinate the entire project and all stages are implemented within the established schedule in accordance with the deadlines. Data scientists, machine learning engineers, software developers and domain experts will be assigned to the project team and work closely together during the process. Principal responsibility of the project manager consists in coordinating the use of resources and ensuring that the work of the team is within the scope of the project and the established schedule, with clarity of communication and actions monitoring.

Financial Analysis

- The financial analysis in the project involves estimation of costs and potential sources of finance. The following are the cost factors that are vital to the project:
- Hardware and Software: (a) For training models, powerful GPUs would be required, and Google Cloud would have to be rented to use the cloud services, and these would require about 100 to 500 dollars. Also, development software licenses like MATLAB, PyTorch and
- Acquisition Costs and Data Preprocessing: Data acquisition cost of IP102 dataset is 0 as it is free, while data augmentation tools, storage and preprocessing have been estimated at 300 USD.
- Deployment Costs: Cost to develop a mobile application and server side infrastructure to deploy the system is estimated to cost approximately 5,00.
- Marketing and Outreach Expenses: Marketing and outreach will cost \$400 to ensure that the system becomes accessible to farmers and this would involve coming up with tutorials and training.

Table 5.1: Cost Analysis and Budget Estimation for Deep Learning-based Pest Detection System

Cost Factor	Estimated Cost (USD)	Alternate Estimated Cost (USD)	Description/Rationale
Hardware and Software	100 - 500	200 - 800	Costs for renting cloud services (e.g., Google Cloud) and purchasing GPUs for training models. Software licenses for

Cost Factor	Estimated Cost (USD)	Alternate Estimated Cost (USD)	Description/Rationale
			tools like MATLAB, PyTorch, etc. Higher cost for more powerful hardware or services.
Data Acquisition and Preprocessing	300	300	No cost for acquiring the IP102 dataset (free). Includes costs for data augmentation tools, storage, and preprocessing operations.
Deployment Costs	500	600	Development of mobile app and server-side infrastructure for deployment. Higher costs if additional features or scaling is required.
Marketing and Outreach	400	500	Costs for marketing the system, creating tutorials, and providing training resources. Larger campaigns may increase costs.
Total Estimated Cost	1700	1600 - 2200	

Potential Funding Sources

1. There are various sources of funding that can be utilized by this project:
2. Government Funding: Government programs of subsidizing agricultural

technologies, particularly among the rural population, can be a source of financing that is needed.

3. Private Sector Sponsorship: partnership with agricultural technology firms could provide sources of funding to develop and deploy.
4. Crowdfunding: people might be interested in agricultural innovation, and thus, crowdfunding can be an alternative to get such funding.
5. Non-profit Organizations: NGOs with agendas in agriculture or poverty reduction will most likely be interested in financing a project that will lead to the positive transformation of the lives of rural farmers.

Return on Investment (ROI)

Enrique Iniguez will be able to determine the ROI of this project in the form of a tangible and an intangible benefit:

- Better Pest Management The new system can significantly enhance agricultural productivity due to management of the pests hence increasing income among farmers due to high crop productivity. As an example, a farmer possessing crops worth 5,000 would receive 500 to 800 dollars additional earnings due to low pest attack.
- These are advancements in agricultural technology, reduced use of pesticides, increased sustainability and security of food.

The ROI will be based on scalability of the system in various regions and adoption rate of the system by the farmers. The general gains that encompass the long-term agricultural output and sustainable agricultural requirements would most likely be more than the cost incurred in executing the project. In conclusion, the investment will be beneficial with respect to the project. The wise choice of funding and the resources, and the effective introduction of the system will render the project successful and sustainable as seen in the long-term perspective.

5.4 Complex Engineering Problem

5.4.1 Complex Problem Solving

Table 5.2: Mapping with Complex Engineering Problem.

EP1 Dept of Knowle dge	EP2 Range Of Conflictin g Requirem ents	EP3 Dept h of Anal ysis	EP4 Familia rity of Issues	EP5 Extent of Applic able Codes	EP6 Extent Of Stake- holder Involve ment	EP7 Interdepen dence
✓	✓	✓	✓	✓	✓	✓

Justifications:

- **EP1: Dept of Knowledge:** The project demanded deep knowledge of computer vision, deep learning techniques, and model architectures like Vision Transformer (ViT) and Data-efficient Image Transformer (DeiT). It required expertise in model fine-tuning, knowledge distillation, and fusion strategies (early and late fusion, majority voting). Proficiency in using frameworks like PyTorch and TensorFlow, along with model evaluation metrics such as accuracy, F1-score, and AUC, was crucial. Familiarity with data preprocessing techniques and handling imbalanced datasets was essential to improve model generalization.
- **EP2: Scope of Competing Requirements:** The system needed to balance high accuracy with computational efficiency. On one hand, it needed to deliver high accuracy in pest classification, especially for diverse pest species. On the other, it had to be optimized for real-time deployment in resource-constrained environments (e.g., mobile devices). The solution had to perform well despite class imbalance and the availability of limited labeled data, and at the same time, support small-scale farmers in remote areas.
- **EP3: Depth of Analysis:** In-depth analysis of various fusion methods (early and late fusion), knowledge distillation, and ensemble learning techniques were conducted to improve model robustness. Performance evaluation involved comparing accuracy, precision, recall, F1-scores, and AUC for multiple models (ViT, DeiT, and EfficientNetB4) in classifying pests. A deep dive into trade-offs between computational cost and model accuracy was carried out, focusing on how each model performed with the IP102 dataset.
- **EP4: Familiarity of Issues:** The project addressed several challenges, such as class imbalance in the IP102 dataset and the computational efficiency required for real-

time pest detection in low-resource settings. Data preprocessing methods like oversampling, augmentation (rotation, flipping, cropping), and normalization were used to overcome data quality issues and ensure the models performed well in diverse environments. Familiarity with these issues allowed the system to handle real-world agricultural conditions effectively.

- **EP5: Scope of Applicable Codes:** The system followed industry-standard frameworks and ethical guidelines for data processing, ensuring compliance with legal and privacy considerations in agricultural data. It adhered to software quality standards (e.g., ISO/IEC 25010) and best practices in AI deployment. Data handling practices ensured that pest images were processed securely, and ethical standards regarding the use of AI in agriculture were upheld to minimize harmful environmental impacts.
- **EP6: Level of Stakeholder Involvement:** The design of the pest detection system considered the needs of farmers, particularly those in resource-constrained environments. The system's objective was to provide farmers with an affordable, mobile-friendly solution for early pest identification, enabling them to take timely action and reduce pesticide usage. The involvement of agricultural experts and small-scale farmers in the system design was critical for ensuring the system's relevance and practicality.
- **EP7: Interdependence:** The project integrated multiple interdependent deep learning components, including ViT and DeiT models for feature extraction, knowledge distillation to transfer knowledge from large models to smaller, more efficient ones, and ensemble methods (early fusion, late fusion, and majority voting) to increase model robustness. The interaction between these modules was essential for creating a scalable and accurate pest detection system that could handle varying pest species and environmental conditions.

Table 5.3: Mapping with knowledge Profile.

K1 Natu ral Scien ce	K2 Mathem atics	K3 Engineeri ng Fundamen tals	K4 Speciali st Knowle dge	K5 Engineer ing Design	K6 Engineer ing Practice	K7 Comprehe nsion	K8 Resear ch Literat ure
		✓	✓	✓	✓		✓

Justifications:

- **K3 - Engineering Fundamentals:** The core principles of engineering were applied in designing and optimizing deep learning models to accurately detect and classify pests. Techniques such as model evaluation metrics (accuracy, precision, recall, F1-score) and optimization strategies (e.g., using Adam optimizer, fine-tuning models) were essential for ensuring the system's performance. These principles guided the adjustments made to improve the accuracy and efficiency of

the models.

- **K4 - Specialist Knowledge:** Specialized knowledge in deep learning, computer vision, and transformer-based models (ViT and DeiT) was critical for the success of this study. Understanding how these models work, as well as techniques like knowledge distillation, early/late fusion, and ensemble methods, allowed the research to propose effective solutions to agricultural pest detection. The integration of these models to achieve high performance and scalability was informed by expert knowledge in AI and machine learning.
- **K5 - Engineering Design:** The engineering design focused on creating a robust, scalable system capable of accurately detecting pests across various agricultural environments. This involved designing a pipeline that incorporates data preprocessing (e.g., resizing, normalization, augmentation), model architecture (ViT, DeiT), and knowledge distillation. The system was built with both functional (accuracy, efficiency) and non-functional (resource constraints, real-time performance) requirements in mind, ensuring its applicability in low-resource agricultural settings.
- **K6 - Engineering Practice:** The project adhered to engineering practices such as ensuring the repeatability and rigor of the research. The methodology included systematic data preprocessing, fine-tuning of pre-trained models, performance evaluation using standardized metrics, and robust reporting of results. These practices ensured that the system could be consistently applied in real-world scenarios and scaled effectively in agricultural systems.
- **K8 - Research Literature:** A thorough review of the literature on pest detection, deep learning models (such as CNNs, ViTs, DeITs), knowledge distillation, and ensemble learning informed the methodology. This review helped identify gaps in existing approaches and guided the development of a more efficient system. The literature also provided insights into the challenges of class imbalance and computational efficiency, which were addressed by incorporating innovative solutions like knowledge distillation and model fusion.

5.4.2 Engineering Activities

The project cycle also reflected a number of sophisticated engineering phases, such as dataset generation, training of the models, testing them, and designing the system.

Complex Engineering Activities Mapping.

This section aims at showing how different activities were integrated into the project lifecycle by mapping the entire issue onto the Engineering Activities (EAs). Creative thinking and close observation of the impacts on the society and the environment also

required a number of resources, stakeholder engagement, and thinking to come up with an ethical invoice data extraction system. The project is a good example of how engineering practice extends beyond technical implementation to broader effects and professional obligations by aligning itself with EA1-EA5.

Table 5.4: Mapping with Complex Engineering Activities.

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

Justifications:

- **EA1 - Variety of Resources:** The project made use of various resources such as the IP102 dataset, high-performance GPUs (NVIDIA RTX 3080 Ti), and deep learning frameworks like TensorFlow and PyTorch. Cloud-based platforms like Google Colab were also utilized for model training, and version control tools like GitHub helped in collaboration and code management. These resources contributed to building and optimizing the pest detection system, showcasing the diversity of tools available for the research.
- **EA2 - Level of Interaction:** The system development involved a range of stakeholders, including domain experts in agriculture, deep learning specialists, and end users such as farmers and agricultural professionals. The iterative process of refining the model with input from various perspectives, including academic advisors and agricultural experts, helped to shape a system that is relevant and practical for real-world applications.
- **EA3 - Innovation:** The innovation in this research lies in the integration of state-of-the-art deep learning models, such as Vision Transformer (ViT) and Data-efficient Image Transformer (DeiT), combined with novel methods like knowledge distillation and ensemble learning. This fusion of models with advanced techniques allowed for efficient pest detection, particularly in low-resource settings, while maintaining high accuracy and computational efficiency.
- **EA4 - Consequences for Society and Environment:** The pest detection system developed in this project has significant positive consequences for society and the environment. By accurately detecting pests early, farmers can reduce pesticide use, leading to more sustainable agricultural practices. This reduction in chemical usage minimizes environmental pollution, conserves biodiversity, and

contributes to healthier ecosystems. Moreover, the system can help small-scale farmers in resource-limited regions, promoting food security and economic growth.

- **EA5 - Familiarity:** The project leveraged prior knowledge from coursework and experience with deep learning models, particularly transformers and ensemble methods. This familiarity allowed for rapid prototyping, experimentation, and fine-tuning of the models. The ability to draw on previous expertise ensured that the system could be developed efficiently, despite limited resources and the complexity of the task.

5.5 Summary

The chapter addresses the engineering standards, design considerations and effect of the deep learn-based pest detection system. It includes compliance to industry standards of software, hardware and communication to achieve efficiency, reliability and scalability. The system uses the Vision Transformer (ViT) and Data efficient Image Transformer (DeiT) architecture, which follows the common practice of machine learning usage in order to perform optimally. The societal impact of the project is strong, especially on smallholder farmers who face resource limited environments and whose livelihoods are affected by GMFs through decreased pesticides and enhanced food safety and health. The energy efficient design of the system makes it be deployed in remote areas on both mobile and edge devices. Measures were taken to avoid harm by taking into account such ethical considerations as fairness, honesty, and confidentiality. Sustainability in the chapter is highlighted through low-cost mobile friendly solutions, interaction with communities, and frequent updates of the system. Lastly, it also points to engineering issues, including the trade-off between model accuracy and computational load and the emphasis on real time performance.

Chapter 6

Conclusion

It is a chapter describing the research work done and outlining it with its main findings, significance, and the contributions of the work, together with limitations and future directions. The chapter concludes the research by making speculations on the whole process and prospects of the deep learning-based pest detection system.

6.1 Summary

This study proposes a deep learning framework solution as a method of classifying pests and representing more accurate pest detection in agriculture. Through use of the IP102 dataset that contains 102 species of pests, the study will limit its discussion to the six so prominent pest species. Vision Transformer (ViT) and Data-efficient Image Transformer (DeiT) models are used in the system and optimised with transfer learning, knowledge distillation and ensemble fusion strategies. The study emphasizes the usefulness of data preprocessing and augmentation strategies to deal with the problem of class imbalance and better generalization of a model. The evaluation values- accuracy, precision, recall, F1-score, and AUC indicate Late Fusion model has the highest accuracy of 98.20 and Teacher Model had the highest precision of 98.34. Although computationally efficient, the Student Model had a slightly smaller accuracy rate of 96.50% in comparison with the Student Model. Besides benefiting farmers, most notably small-scale ones in resource-limited contexts, with a mobile-compatible method of pest detection, the system propagates sustainability to the economy by minimizing the use of pesticides, making food more safe and the environment more viable. Ethical considerations such as fairness, affordability, data privacy have been put into its design such that the technology is useful to agricultural communities with minimal environmental degradation. The final result of this study is more cost, time, and ecologically effective pest control in agriculture.

6.2 Limitation

Although the study introduces an in-depth deep learning-based pest detection system, one should not disregard some limitations. To begin with, the IP102 dataset is a big one yet it covers few pest species; hence, it limits the general applicability of the model utilizing it. Also, the data may not adequately represent the cross-environmental appearances of the pests given the varying environmental conditions in the real sense lighting, weather, crop environments and so on. Although data augmentation was used to resolve the class imbalance, it might be not enough to cover every possible variation to the pest species. Moreover, although the models are optimized using transfer learning

and knowledge distillation, they remain memory heavy, especially when they are to be applied during real-time in the resource-constrained deployment. Despite the high computation efficiency of the Student Model, the percentage of its accuracy (96.50) was less than that of Teacher Model (98.34), which might impact its reliability in such important applications as detection of pests. Dependency on pre-trained models such as ViT and DeiT can be another factor that makes the system not very flexible since it might not work well on data that is vastly dissimilar to that of ImageNet. Lastly, though the system may be mobile-friendly, the execution may also be inconvenienced by connection problems or the availability to requisite hardware, which may curtail its extensive use among small-scale farmers at the rural and remote locations.

6.3 Future Work

There are several issues with the deep learning pest-detecting system that would be improved with more development. Firstly, the models' generalizability would have been improved, by the larger scope of the identity of the pest species within the data set, or by more diversity in the values of the environmental variables. In fact, more information about the real conditions in different geographical range and crops would result in a system more adapted to different calibres of farming. In addition, other sophisticated data augmentation techniques to generate synthetic images such as Generative Adversarial Networks (GANs) must be attempted in order to overcome the curse of class imbalance and introduce more variability in the dataset. The second area where one would like to push this aspect even more is the polishing of the models to tools which would be further towards the edge of the system e.g. considering concepts around model pruning, quantization, distillation, to fit the models to edge devices, i.e., smartphones or embedded systems, and aim even accuracy beyond the efficiency of the resource-constrained setup. The future scope can be in real-time detection and system integration in the context of smarter farming where automatic systems and pest detection systems can also be integrated with IoT-based technologies, which can be utilized for real-time pest control. Strengthening the user interface that is more rural and the farmers can understand and the multilingual knowledge support in the rural and remote areas would facilitate the use and access of the system by the farmers.

Moreover, the establishment of interconnection between the pest detection system and the rest of the monitors in the agricultural sphere, i.e., crop health, weather forecast, etc., could potentially provide a farmer with a systematic approach to crop productivity maintenance and losses reduction. Finally, being able to keep track of the overall performance throughout the history and modify the model in accordance to the feedback given by farmers and new data obtained would also aid in keeping the system efficient and responsive to the issues at hand in the agricultural sector.

References

- [1] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” *International journal of computer vision*, vol. 59, no. 2, pp. 167–181, 2004.
- [2] A. Vedaldi and S. Soatto, “Quick shift and kernel methods for mode seeking,” in *European conference on computer vision*, Springer, 2008, pp. 705–718.
- [3] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “Slic superpixels compared to state-of-the-art superpixel methods,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.
- [4] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [5] Gaston, K. J., “The Magnitude of Global Insect Species Richness,” *Conserv. Biol.* 2010, 5, 283–296.
- [6] Martineau M, Conte D, Raveaux R, Arnault I, Munier D, Venturini G. A survey on image-based insect classification. *Pattern Recognit.* 2017; 65. <https://doi.org/10.1016/j.patcog.2016.12.020>.
- [7] Ghiasi, G. et al. Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation. Preprint at arXiv:2012.07177 (2020).
- [8] Johnny L Miranda, B Gerado, Bartolome T Tanguil, “Pest Detection and Extraction Using Image Processing Techniques” *International Journal of communication and Engineering*, DOI:10.7763/IJCCE.2014.V3.317 Corpus ID: 8891485.
- [9] Chen, J., Wu, J., Wang, Z., Qiang, H., Cai, G., Tan, C., Zhao, C., 2021d. Detecting ripe fruits under natural occlusion and illumination conditions. *Comput. Electron. Agric.* 190.
- [10] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, “Gather-excite: Exploiting feature context in convolutional neural networks,” *arXiv preprint arXiv:1810.12348*, 2018.
- [11] M.E. Karar, “Robust RBF neural network-based backstepping controller for implantable cardiac pacemakers,” *Int. J. Adapt Control Signal Process.* 32 (2018) 1040–1051.
- [12] X. Wu, C. Zhan, Y.-K. Lai, M.-M. Cheng, and J. Yang, “Ip102: A large scale benchmark dataset for insect pest recognition,” in *Proceedings of the IEEE/CVF*

- Conference on Computer Vision and Pattern Recognition, 2019, pp. 8787–8796.
- [13] F. Ren, W. Liu, and G. Wu, “Feature reuse residual networks for insect pest recognition,” *IEEE Access*, vol. 7, pp. 122758–122768, 2019.
 - [14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
 - [15] T. Kasinathan and S. R. Uyyala, “Machine learning ensemble with image processing for pest identification and classification in field crops,” *Neural Computing and Applications*, pp. 1–14, 2021.
 - [16] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International Conference on Machine Learning*, PMLR, 2019, pp. 6105–6114.
 - [17] L. Nanni, A. Manfe, G. Maguolo, A. Lumini, and S. Brahnam, “High performing ensemble of convolutional neural networks for insect pest image detection,” *arXiv preprint arXiv:2108.12539*, 2021.
 - [18] L. Nanni, G. Maguolo, and F. Pancino, “Insect pest image detection and recognition based on bio-inspired methods,” *Ecological Informatics*, vol. 57, p. 101089, 2020.
 - [19] Pereira, V.J.; da Cunha, J.P.A.R.; de Moraes, T.P.; Ribeiro-Oliveira, J.P.; de Moraes, J.B. Physical-chemical properties of pesticides: Concepts, applications, and interactions with the environment. *Biosci. J.* 2016, 32, 627–641. [CrossRef].
 - [20] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
 - [21] Y. Liu, L. Wang, J. Cheng, C. Li, and X. Chen, “Multi-focus image fusion: A survey of the state of the art,” *Information Fusion*, vol. 64, pp. 71–91, 2020.
 - [22] Xu C, Yu C, Zhang S, Wang X,” Multi-Scale Convolution-Capsule Network for Crop Insect Pest Recognition,” *Electronics* 2022,11, 1630. [CrossRef].
 - [23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, “SSD: Single Shot MultiBox Detector,” in: B.Leibe, J. Matas, N. Sebe, M. Welling (Eds.), *Computer Vision – ECCV 2016*, Springer International Publishing, Cham, 2016, pp. 21–37.[CrossRef].

- [24] Mohammed Esmail Karar, Alsunaydi F, Albusaymi S, Alotaibi S, "A New Mobile Application of Agricultural Pests Recognition Using Deep Learning in Cloud Computing System," *Alex. Eng. J.* 2021, 60, 4423–4432. [CrossRef].
- [25] Nour Khalifa, N.E, Loey M, Taha M, "Insect Pests Recognition Based on Deep Transfer Learning Models," *J. Theor. Appl. Inf. Technol.* 2020, 98, 60–68. [CrossRef].
- [26] S.-H. Tsang, "Review: Densenet—dense convolutional network (image classification)," en línea.[consulta: 7 abril 2019]. Disponible en: <https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803>, 2018.
- [27] J. Wäldchen and P. Mäder, "Machine learning for image based species identification," *Methods in Ecology and Evolution*, vol. 9, no. 11, pp. 2216–2225, 2018.
- [28] I. P. De Sousa, M. M. B. R. Vellasco, and E. C. Da Silva, "Local interpretable model-agnostic explanations for classification of lymph node metastases," *Sensors (Basel, Switzerland)*, vol. 19, no. 13, 2019.

ORIGINALITY REPORT

20%

SIMILARITY INDEX

12%

INTERNET SOURCES

15%

PUBLICATIONS

7%

STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	2%
2	Submitted to Daffodil International University Student Paper	2%
3	dspace.bracu.ac.bd Internet Source	1%
4	Niranjan C Kundur, P B Mallikarjuna. "Insect Pest Image Detection and Classification using Deep Learning", International Journal of Advanced Computer Science and Applications, 2022 Publication	1%
5	arxiv.org Internet Source	1%
6	S.P. Jani, M. Adam Khan. "Applications of AI in Smart Technologies and Manufacturing", CRC Press, 2025 Publication	1%
7	Roberto Castro, Israel Pineda, Wansu Lim, Manuel Eugenio Morocho-Cayamcela. "Deep Learning Approaches Based on Transformer Architectures for Image Captioning Tasks", IEEE Access, 2022 Publication	<1%
8	www.preprints.org Internet Source	<1%