A DEEP LEARNING MODEL FOR DECISION SUPPORT OF STAKEHOLDERS IN CROP DISEASES AND PEST MANAGEMENT

Vincent Mbandu Ochango

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Information Technology of Murang'a University of Technology

August, 2026

DECLARATION

Student Declaration

I hereby declare that this thesis is my original work and to the best of my knowledge has not been presented for a degree awarded in this or any other university. I understand what plagiarism is and I am aware of the University policy in this regard. I understand that any false claim in respect of this work shall result in disciplinary action in accordance with the University Anti-Plagiarism Policy.

_____ _____

Vincent Mbandu Ochango Date

SC500/5807/2022

Supervisors' Approval

The undersigned certify that they have read and hereby recommend for acceptance of Murang'a University of Technology thesis entitled "**A Deep Learning Model for Decision Support of Stakeholders in Crop Diseases and Pest Management.**"

_____ _____

Dr. Geoffrey Mariga Wambugu, Date

Department of Information Technonology,

Murang'a University of Technology.

_____ _____

Dr. Aaron Mogeni Oirere, Date

Department of Computer Science,

Murang'a University of Technology.

DEDICATION

Dedicated to:

My mother Sarah Agiso Ochango,

Throughout my Degree of Doctor of Philosophy in Information Technology, you have been my rock. I was able to complete my studies with the financial and spiritual assistance you gave me. You inspired me to keep going no matter how tough things became every time I felt like giving up. May the Almighty God bless you.

My siblings Caroline Atemi Ochango and David Atsieli,

Your unending care for my development and the sound counsel you offered helped me finish my thesis on time; for that, I am eternally grateful, and I pray that God showers his plentiful blessings on you.

My uncle Benson Amukoa Abwao,

In addition to the financial aid you gave me, you showed genuine interest in how I was doing at whatever opportunity we had to meet. My whole family and I will benefit from the completion of my degree, as you have consistently emphasized. You remain in my prayers.

And

In honor of my late father, Clyde Apeli Ochango,

I will never forget the words of encouragement you shared with me.

ACKNOWLEDGMENT

I am grateful to the Almighty God for blessing me with the physical and mental fortitude to finish my thesis on schedule.

I express my deepest appreciation to my thesis supervisors, Dr. Geoffrey Mariga Wambugu and Dr. Aaron Mogeni Oirere, who helped me immensely during the whole research process. My exploration of strong ideas and development of a well-written thesis were both greatly aided by the extensive conversation and time we spent together. I took comments positively which enabled me to answer all of my supervisors' criticisms, which helped me enhance my work and produce a well-documented thesis. In addition, my supervisors were always accessible whenever I had questions or wanted clarification, and they graciously allowed me the time I needed to implement their suggestions.

My friend Dr. James Mutinda has been an incredible source of support for me, and I am very grateful. Not only that, but you have taken the time to teach me how to structure my thesis, and whenever I have asked you to evaluate my work, you have been there to listen and provide feedback.

Also, I want to express my gratitude to all my classmates with whom I began this Degree of Doctor of Philosophy in Information Technology. I know that we had a tough journey ahead of us, but your support and inspiration kept me going, and I pray that God bless all of you.

I would also want to express my gratitude for the many online resources that were instrumental in my ability to complete my thesis on schedule.

MAY GOD BLESS ALL OF YOU

ABSTRACT

Crop diseases and pest infestations pose a serious threat to food security and the livelihoods of farmers across the world, posing substantial difficulties to the agricultural industry. Stakeholders, such as farmers, agronomists, and agricultural extension officers, need decision support systems that are both accurate and fast to handle these difficulties. A deep learning model tailored to decision support in crop disease and pest management was the intended outcome of this study. The goals were to create a hybrid model for correct crop diseases and pest classification using a deep neural network, create an easy-to-use software platform, validate the model's generalizability in different agricultural contexts, and identify and analyze important factors affecting decision support. The first objective of the research included a thorough examination of the main features of deep learning models that influence decision support for crop diseases and pest management. The collection of high-resolution, labelled datasets of crop images from the Kaggle website was done with an emphasis on data quality. These datasets represented diverse forms of crop diseases and pest infestations. In order to increase the classification accuracy, a hybrid model was built that combined a Convolution Neural Network (CNN) with Support Vector Machines (SVM). This model leveraged the capabilities of both techniques. Stakeholders were able to upload images of affected crops and get prompt diagnoses and recommendations when the model was integrated into an intuitive software platform. Extensive testing was conducted across several datasets, including tomatoes, maize, and pests, to assess the model's robustness and generalizability. On the maize data set, the hybrid CNN-SVM model achieved 97.13% accuracy, on the tomato dataset, it achieved 97.04% accuracy, while on the pest dataset, it achieved 97.02% accuracy. Analyses comparing DenseNet, AlexNet, EfficientNet, and VGG showed that their accuracy rates were lower; DenseNet achieved 96.99% for maize, 92.60% for tomatoes, and 96.94% for pest classification. Stakeholders were able to improve their capacity for proactive pest and disease management in crops through the use of the user-friendly software platform's proactive decision support. Data quality, efficient feature extraction, model design, and environmental conditions consideration were some of the important factors emphasized by the study as impacting decision support in crop diseases and pest management. In order to get reliable predictions, the results highlighted the need to use high-quality training data and complex model architectures. The study proved that deep learning models greatly enhanced agricultural decision-making by addressing problems related to crop diseases and pest management. When it came to decision support for stakeholders dealing with pest control and crop diseases, the developed hybrid model and its incorporation into an intuitive software platform provided substantial improvements. The hybrid model, according to the research, improved agricultural sustainability and production by providing correct insights in a timely manner. To achieve broad acceptance and influence in the agricultural sector, future work should attempt to enhance the software platform, validate the model's performance across several agricultural settings, and increase the datasets. The findings and methods presented here transformed crop diseases and pest management by providing useful information for decision support systems in the agricultural sector.

TABLE OF CONTENTS

DECLARATION ..... **ii**
DEDICATION ..... **iii**
ACKNOWLEDGMENT ..... **iv**
ABSTRACT ..... **v**
LIST OF TABLES ..... **ix**
LIST OF FIGURES ..... **x**
ACRONYMS AND ABBREVIATIONS ..... **xii**
CHAPTER ONE: INTRODUCTION **1**
1.1 Background of the Study 1
1.2 Statement of the Problem 4
1.3 Research Objectives 6
1.3.1 Main Objective 6
1.3.2 Specific Objectives 6
1.4 Research Questions 6
1.5 Significance of the Study 7
1.6 Scope of the Study 8
1.7 Limitations of the Study 10
1.8 Contributions of the Thesis 12
1.9 Organization of the Thesis 13
CHAPTER TWO: LITERATURE REVIEW **15**
2.1 Introduction 15
2.2 Theoretical Framework 16
2.2.1 Decision Support Systems Theory 16
2.2.2 Classification Theory 17
2.2.3 Technology Acceptance Model 19
2.2.4 Generalizability Theory 20
2.3 Factors that Influence Decision Support in Crop Diseases and Pest Management 23
2.4 Explainability and Trust 25
2.5 Deep Neural Network Model Development Approaches 27
2.6 Model Implementation Approaches 48
2.7 Model Validation Approaches 51
2.8 Research Gaps 54
2.9 Summary 55
CHAPTER THREE: METHODOLOGY **57**
3.1 Introduction 57
3.2 Research Philosophy 57

3.3 Research Design 58
3.4 Research Process 59
3.5 Description of the Training and Test Data Set 61
3.6 Experimental Setup 67
3.7 Data Preprocessing 68
3.8 Loading the Dataset 69
3.9 Identification and Analysis of Deep Learning Model Factors Influencing Decision Support in Crop Diseases and Pest Management 70
3.10 Development of an Enhanced Deep Neural Network Model 77
3.10.1 Convolution Neural Network 81
3.10.2 The Softmax Layer 89
3.10.3 The Support Vector Machine 90
3.11 Software Platform Implementation and Integration 97
3.12 Model Validation 101
3.12.1 Classification Accuracy 102
3.12.2 Precision 102
3.12.3 Recall 102
3.12.4 F1 Score 102
3.12.5 Training and Validation Loss 103
3.13 Ethical Considerations 104
3.14 Summary 105
CHAPTER FOUR: DEEP LEARNING MODEL FACTORS INFLUENCING DECISION SUPPORT IN CROP DISEASES AND PEST MANAGEMENT 106
4.1 Introduction 106
4.2 DenseNet 109
4.3 AlexNet 112
4.4 EfficientNet 117
4.5 Visual Geometry Group 121
4.6 Convolution Neural Network 124
4.7 Summary 130
CHAPTER FIVE: A HYBRID DEEP LEARNING MODEL FOR DECISION SUPPORT IN CROP DISEASES AND PEST MANAGEMENT 131
5.1 Introduction 131
5.2 Deep Learning Model 131
5.3 Convolution Layer 132
5.4 Activation Function 133
5.5 Pooling Layer 135
5.6 Fully Connected Layer 136

5.7 Feature Vector Generation 137
5.8 Support Vector Machine Classifier 138
5.9 Training the Hybrid Model 140
5.10 Model Evaluation 148
5.11 Summary 151
CHAPTER SIX: MODEL IMPLEMENTATION AND APPLICATION INTEGRATION 153
6.1 Introduction 153
6.2 Deployment Strategy 154
6.3 Integration with the Application 155
6.4 Summary 163
CHAPTER SEVEN: VALIDATION OF MODEL ROBUSTNESS AND GENERALIZABILITY 165
7.1 Introduction 165
7.2 Validation Methodology 166
7.3 Generalizability Across Diverse Crop Diseases and Pest Variations 167
7.4 Real World Agricultural Setting Validation 167
7.5 Comparative Analysis with Existing Models 177
7.5.1 Confusion Matrix 181
7.5.2 Train and Validation Loss 187
7.5.3 Test of Significance 193
7.6 Summary 198
CHAPTER EIGHT: SUMMARY, CONCLUSION AND RECOMMENDATIONS 200
8.1 Summary 200
8.2 Conclusion 201
8.2.1 Deep Learning Model Factors 202
8.2.2 Enhanced Model 203
8.2.3 Implementation of a User-Friendly Platform 203
8.3 Recommendations 204
8.3.1 Enhancing the Model's Capabilities 205
8.3.2 Exploring Other Hybrid Approaches 205
8.3.3 Improving the Software Platform 205
8.3.4 Further Validation of Robustness and Generalizability 206
8.4 Recommendations for Future Research 207
REFERENCES 210
APPENDICES 243

LIST OF TABLES

Table 2.1 Deep Learning Model Factors 25
Table 2.2 Summary of the Models Limitations 38
Table 2.3 Deep Learning Model Implementation Approaches 50
Table 2.4 Model Validation Approaches 53
Table 4.1 DenseNet Architectural Components. 110
Table 4.2 AlexNet Architectural Components. 114
Table 4.3 EfficientNet Architectural Components. 119
Table 4.4 VGG Architectural Components. 122
Table 4.5 CNN Architectural Components. 126
Table 7.1 Classification Accuracy of Various Deep Learning Models 194

LIST OF FIGURES

Figure 2.1 Theoretical Framework for Deep Learning Based Decision Support 21
Figure 3.1 Research Process 60
Figure 3.2 Sample of Maize Leaf Disease Images 64
Figure 3.3 Sample of Pests Images 66
Figure 3.4 Systematic Literature Review on Deep Learning Model Factors Influencing Decision Support in Crop Diseases and Pest Management 72
Figure 3.5 Article Selection Process 74
Figure 3.6 CNN-SVM Model 80
Figure 3.7 Max Pooling 86
Figure 3.8 Pooling Operation 87
Figure 3.9 Softmax Layer 90
Figure 3.10 CNN-SVM Algorithm 96
Figure 3.11 Implementation of a Deep Learning Model in a Web-Based Decision Support System 100
Figure 3.12: Confusion Matrix 101
Figure 5.1 CNN-SVM Model 132
Figure 5.2 Schematic Diagram for Crops and Pests Classification Model 142
Figure 6.1 User Interface for the Application 156
Figure 6.2 Upload Image 157
Figure 6.3 Image Analysis for Maize Diseases 158
Figure 6.4 Find Agrovets 159
Figure 6.5 Display of Agrovets Around Thika Town 161
Figure 6.6 Plant Expert 162
Figure 7.1 User Interface for the Application 169
Figure 7.2 Upload Image 170
Figure 7.3 Image Analysis for Maize Diseases 171
Figure 7.4 Image Analysis for Pest 172
Figure 7.5 Find Agrovets 174
Figure 7.6 Display of Agrovets Around Thika Town 175
Figure 7.7 Plant Expert 176
Figure 7.8 Accuracy of Various Deep Learning Models 179
Figure 7.9 Confusion Matrix for Densenet on Maize Dataset 182
Figure 7.10 Confusion Matrix for Densenet on Pest Dataset 184
Figure 7.11 Confusion Matrix for the Hybrid Model (CNN + SVM) on Maize Dataset 186
Figure 7.12 Train and Test Loss for Visual Geometry Group on Tomato Dataset .. 189

Figure 7.13 Train and Test Loss for the Hybrid of CNN + SVM on Tomato Dataset 191

ACRONYMS AND ABBREVIATIONS

ANN	Artificial Neural Network
DT	Decision Tree
DL	Deep Learning
IEEE	Institute of Electrical and Electronics Engineers
KNN	K-Nearest Neighbour
LBP	Local Binary Pattern
NB	Naïve Bayesian
RF	Random Forest
SVC	Support Vector Classifier
SVM	Support Vector Machine
CNN	Convolution Neural Network
ML	Machine Learning
DL	Deep Learning
CONV2D	2D Convolution Layer
RNN	Recurrent Neural Network
AI	Artificial Intelligence
GPU	Graphics Processing Unit
SGD	Stochastic Gradient Descent
BCE	Binary Cross-Entropy
ReLU	Rectified Linear Unit
DNN	Deep Neural Network
RL	Reinforcement Learning
TPUs	Tensor Processing Units
TL	Transfer Learning
VGG	Visual Geometry Group
DenseNet	Dense Convolutional Network

CHAPTER ONE
INTRODUCTION

1.1 Background of the Study

The agricultural industry, which supplies food, jobs, and raw materials to other industries, has maintained its position as one of the world's most important economic drivers. But crop diseases and pests have long been major problems for agricultural output; research acknowledges that these factors account for 20% to 40% of yearly crop production losses worldwide [1]. Agricultural systems are under increasing pressure to implement new technology that can boost productivity and sustainability to meet the rising demand for food caused by the world's increasing population [2]. Manual inspection and expert consultations were the backbone of traditional crop disease detection and pest control approaches, but they were inefficient, subjective, and difficult to use, particularly in rural locations [3]. As a result, artificial intelligence and deep learning in particular have become a game changer in the agricultural sector, allowing for the automatic and accurate identification of crop diseases and pests through image classification [1], [2].

Convolutional Neural Networks (CNNs) and other deep learning models have shown outstanding performance in identifying pests and classifying crop diseases in the global research landscape [5]. Research from nations like China and India has shown that using artificial intelligence in farming increased early detection and intervention, reduced crop losses, and enhanced farmer decision-making [6]. Existing models often encountered difficulties with scalability, robustness, and generalizability across diverse agro-ecological zones, despite these advancements [7]. In addition, most Decision Support Systems (DSS) that have been deployed globally do not fully integrate deep learning techniques, which hinders their capacity to provide stakeholders with context-aware insights in real-time. This highlighted the need for hybrid solutions that can accurately classify crop diseases and pests while also improving the ability of agricultural stakeholders, including farmers, extension officers to make informed decisions [9].

More than 60% of the African population relies on agriculture for their living, making it the backbone of most economies in the region [10]. Nevertheless, new crop diseases

and pests pose significant threats to food security in Africa, including the fall armyworm (Spodoptera frugiperda), maize lethal necrosis, and cassava mosaic virus [11]. Because of their lack of access to contemporary diagnostic tools and agricultural experts, small-scale farmers are already at a high risk of experiencing significant output losses [12]. Despite promising results, few African farmers have used Information and Communication Technology (ICT) solutions in their operations owing to a lack of understanding and funding [13].

The small-scale farmers are the backbone of Kenya's agricultural sector, which accounts for almost 40% of the total population and 33% of the country's Gross Domestic Product [14]. Crop diseases like northern leaf blight and common rust, as well as pests like fall armyworm, have a devastating impact on maize output, which is a staple crop for most families [15]. Fall armyworm is shown to be responsible for maize losses ranging from 8 to 21 percent per year, which poses a challenge to both food security and farmers' income [16]. There are still a lot of farmers in Kenya who don't have access to timely, accurate, and cost-effective ways to identify crop diseases and pests, even though the government and research institutes have promoted Integrated Pest Management (IPM) strategies and extension services [17]. This demonstrates the critical need to develop new tools that make use of deep learning technology to improve decision support for various stakeholders in the food production value chain [18].

The development of a decision support system based on deep learning was, therefore, both essential and applicable. This system helped farmers and other stakeholders in Kenya stay informed about crop diseases and pests in real-time. It also recommended treatments based on the information, hence encouraging sustainable agricultural practices. This innovation helped stakeholders in Kenya to increase productivity, boost food security, and make agricultural data-driven decisions [3], [4].

To make informed decisions about crop diseases and pest management, agricultural stakeholders-farmers, agricultural extension agents, and legislators-need timely and reliable information. Their capacity to efficiently control crop health may be greatly improved by the availability of trustworthy decision support models. By enabling stakeholders to take proactive steps to mitigate damage and reduce the use of chemical pesticides, these decision support models can offer real-time insights into the presence

and severity of crop diseases and pest infestations. This will promote sustainable and environmentally friendly agriculture[5], [6].

The goal of this research was to create a strong decision support model for crop disease and pest control by using the capabilities of deep learning. The model will be made to reliably detect the presence of pests and illnesses by processing images of crops and pests[7], [8]. By doing this, it will provide farmers and other stakeholders with up-to-date information that will enable them to reduce crop losses, maximize resource allocation, and make educated choices about disease and pest management. In order to guarantee that a variety of stakeholders may benefit from cutting-edge technology, the research will also investigate the viability of implementing this model in actual agricultural settings utilizing accessible technology, such as mobile phones[9], [10], [11].

While deep learning models have shown promise in this area, they cannot function properly without comprehensive training datasets. The generation of varied training samples by the application of multiple changes to the original data, known as data augmentation, emerges as a crucial component in this process. This augmentation will boost the model's capacity to generalize to varied situations and ensure it is more tolerant to variances found in real-world agricultural settings [5], [12], [13].

By emphasizing important patterns in the input data, feature extraction plays a crucial role in deep learning models. To correctly identify symptoms and patterns associated with different crop diseases, extracting discriminative features is crucial in the context of crop diseases and pest control[14], [15], [16]. This process helped the model convert unprocessed input, such as pictures, into useful knowledge for making decisions. With proper feature extraction, stakeholders will learn the reasoning behind the model's suggestions and have more faith in the model's conclusions[17], [18], [19].

In order to overcome the shortcomings of conventional methods, data augmentation and feature extraction will be combined. These methods will boost the overall performance and dependability of deep learning models by enhancing the model's capacity to generalize and identify meaningful information from the testing dataset[20], [21], [22]. The findings of this research will aid in the development of a decision support model for agricultural stakeholders by providing them with clearer

and more useful information to combat crop diseases and pests in a timely manner [7], [23], [24].

The research aimed to solve a significant problem in agriculture by using a deep learning technique to create a decision support model for managing crop diseases and pests[25], [26], [27]. This approach had the capacity to improve agricultural sustainability, lessen the negative environmental effects of pest management methods, and increase global food security by giving stakeholders fast and accurate information[28], [29], [30]. Thus, the research offered a potential remedy for a vital problem by bridging the gap between state-of-the-art technology and the real-world requirements of agricultural stakeholders[31], [32], [33].

1.2 Statement of the Problem

The underlying problem is the significant and ongoing difficulty in controlling pest infestations and crop diseases in agriculture, which results in inaccurate decisions. As a result, it's critical to identify and treat pest infestations and crop diseases as soon as possible. The sustainability of the environment, economic stability, and global food security are all significantly impacted by this issue in a developing country like Kenya. In addition to lowering agricultural product quality and causing significant output losses, crop diseases and pests necessitate the use of chemical pesticides, which can have harmful impacts on the environment and public health.

Conventional techniques for identifying and controlling pests and agricultural diseases often depend on human observation and experience. It's possible that farmers and agricultural extension agents lack the skills or access to the most recent data and tools needed to recognize these problems in a timely and accurate manner. As a result, they can find it difficult to decide how best to apply pesticides, put control measures in place, or distribute resources.

Haque et al. [34] developed a model to examine crop images and detect the presence of pests and illnesses using convolutional neural networks. This technique do have limitations, first and foremost, it needs to have access to vast and varied image collections covering a range of crop varieties, diseases, and pest species for need to give accurate decisions. Furthermore, its generalization was limited since when the model was trained on a particular collection of crops and illnesses, it could not

function effectively with crops or diseases it has never been subjected before. The model's use in various agricultural situations was limited by this generalization problem.

Recognizing transfer learning's ability to use information from pre-trained models, Yong et al [6] adopted it as a strategy to improve decision support in crop disease and pest management. The goal of the author was to enhance the deep learning model's performance in the face of limited labeled data in the agricultural environment by employing a pre-existing model trained on a comparable task to speed up the learning process. The research acknowledged some of the limitations of transfer learning, however, including the domain shift issue and the possibility of biases inherited from the source domain. These difficulties highlighted the need for careful fine-tuning of the transferred model on local or domain-specific data to ensure its adaptability to the various agricultural conditions and to reduce any biases that might affect the dependability of decision support for stakeholders in crop management.

Bajait et al. [6] developed deep learning models for decision support in crop disease and pest control and used convolution Neural Network as a feature extractor. To effectively detect visual clues linked with agricultural illnesses and pest infestations, CNNs must automatically learn hierarchical representations of characteristics from picture data. However, CNNs' inherent focus on local patterns and fixed-size receptive fields was a barrier when it comes to collecting complex characteristics. Capturing subtle and complex aspects was difficult in agricultural settings due to the significant variation in the presentation of diseases and pests. The rigidity of CNNs to process data at varying granularities means that they miss important information that would have improved diagnosis and decision-making. This limitation must be overcome by the researcher if deep learning models are to accurately reflect the nuances of crop diseases and pests and thereby aid stakeholders with more trustworthy and accurate decision-making when confronted with agricultural management issues.

1.3 Research Objectives

1.3.1 Main Objective

To develop and evaluate a deep learning model for decision support of stakeholders in crop diseases and pest management

1.3.2 Specific Objectives

To achieve the main objective, the research will be guided by the following specific objectives:

i. To identify and analyze key factors influencing decision support in crop diseases and pest management using deep learning models.
ii. To develop an enhanced deep neural network model based on the identified factors that accurately identifies and classifies crop diseases and pest infestation for decision support.
iii. To implement the developed deep learning model into a user-friendly software platform providing timely and actionable decision support.
iv. To validate the robustness and generalizability of the implemented model across diverse crop disease types and pest variations in real-world agricultural settings.

1.4 Research Questions

i. What are the key factors influencing decision support in crop diseases and pest management using deep learning models?
ii. How do you develop an enhanced deep neural network model based on the identified factors that accurately identify and classify crop diseases and pest infestation for decision support?
iii. How do you implement the developed deep learning model into a user-friendly software platform providing timely and actionable decision support?
iv. How do you validate the robustness and generalizability of the implemented model across diverse crop disease types and pest variations in real-world agricultural settings?

1.5 Significance of the Study

The research made a contribution by using deep learning to help farmers with important decisions about crop disease and pest control in a way that was accurate, quick, and practical. To better understand how to enhance deep learning models for improved classification accuracy, interpretability, and practicality, the research identified and analyzed critical factors impacting decision support in crop diseases and pest control. Improving deep learning models' effectiveness in agriculture sheds light on data pretreatment methods, model architecture selection, and domain knowledge integration[35], [36]. Traditional techniques of crop disease and pest identification were also discussed in the research, along with their drawbacks, such as being labor-intensive, time-consuming, and inaccurate because of human error.

Improving early detection and response mechanisms was a major goal of developing a deep neural network model that could correctly identify and classify pest infestations and crop diseases. The study found that by integrating Convolution Neural Network (CNN) with Support Vector Machine (SVM), a hybrid deep learning approach was able to improve the model's capacity to differentiate between pests and visually similar crop diseases, as well as increase classification precision and reduce false detection rates. Reduced crop losses, increased food security, and better crop disease and pest management decisions were all made possible by this research. The research also helped advance machine learning applications in agriculture by showing how deep learning models can be improved to identify and classify crop diseases and pests for decision support of stakeholders.

There were major effects on accessibility and usability by integrating the developed model into an intuitive software platform. The study made sure that the system's user-friendly online interface could be readily used by agricultural extension officials and farmers. Users were able to upload images, get real-time classification results, and have access to recommended insecticides, fungicides, and treatment plans. This method improved precision agriculture by lowering the bar for predicting crop health management and paving the way for data-driven decisions. Stakeholders were also able to find local agro-veterinary service providers because of the platform's

geolocation capabilities, which improved both response times and availability of agricultural inputs.

Verifying the model's robustness and applicability across several crop diseases and pest variants guaranteed the solution's dependability for practical agricultural uses. The research offered empirical proof that the model works outside of controlled experimental settings by assessing its performance across various datasets, climatic conditions, and crop types. For Artificial Intelligence driven agriculture solutions to gain confidence and be widely adopted, this validation was vital. Soil health assessment, production prediction, and climate-resilient farming techniques are just a few of the additional agricultural problems that might benefit from the use of deep learning, which was the subject of future research.

The research played a significant role in bridging the gap between theoretical advances in deep learning and their real-world agricultural applications. To increase crop production and sustainability, the study offered a scalable solution by creating a decision support system that is resilient, interpretable, and deployable. This system can be used in many agricultural locations. By showing how Artificial Intelligence-driven decision support systems can revolutionize conventional farming practices, the results enriched agricultural informatics as a whole. Researchers in the field of agriculture and stakeholders were able to use the study's findings to inform data-driven policies and technologies that would improve food security on a global scale [6], [37].

1.6 Scope of the Study

The research's primary objective was to develop a deep-learning model to aid in the identification and classification of crop diseases and pests. The research's stated goal was to enhance agricultural decision-making by the use of more precise, efficient, and accurate deep learning methods for the detection and classification of crop diseases and pest infestations. Data quality, model interpretability, model architecture, model components, feature extraction, and real-time application were some of the parameters that were analyzed in the research. The research looked at how these factors impacted deep learning models' performance and offered ways to optimize them for better classification accuracy and usability. Using Convolution Neural Network (CNN) and Support Vector Machine (SVM) to increase accuracy and reduce misclassification

rates, the research mainly focused on image-based crop diseases and pests classification[38], [39].

The research led to the development of a more advanced deep neural network model that accurately identified and classified crop diseases and pests. By combining Convolution Neural Network (CNN) with Support Vector Machine (SVM), the model's architecture aimed to improve feature extraction and classification. Data preprocessing, augmentation methods, model training, hyperparameter tuning, and performance assessment were all part of the development scope. The research focused on a few crops (maize and tomato) and pests (ants, snails, earwigs, slugs, weevils, wasps, beetles, earthworms, moths, bees, caterpillars, and grasshoppers) which guaranteed that the dataset accurately reflected the wide range of crop disease symptoms and pest types. Nevertheless, to test the performance of the models in a controlled and scalable way, the research only used sample datasets and did not include every potential crop species or environmental condition[40], [41].

Integrating the developed model into an intuitive software platform was an important aspect of the research. Researchers, agricultural officials, and farmers used the platform's web-based decision support system to upload images, get classification results, and see recommended treatments. The software's practical demands were ensured by examining its usability, accessibility, and functionality in the research. Users also made use of the platform's geolocation technologies to locate agro-veterinary service providers in their nearest location. The major objective of the study was to test the efficacy of the deep learning model inside a web-based environment; however, the emphasis was not on building mobile apps or integrating Internet of Things (IoT) technologies[42], [43].

The research further tested the proposed model's performance across varied datasets, climatic circumstances, and crop disease changes to determine its robustness and generalizability. Various deep learning models, such as DenseNet, AlexNet, EfficientNet, VGG, and standard alone CNN architectures, were compared during validation. Although the study's stated goal was to guarantee the model's generalizability to actual agricultural contexts, it failed to thoroughly investigate cybersecurity vulnerabilities, adversarial assaults, or ethical issues with AI-driven agricultural decision-making. Rather than addressing more systemic issues like yield

prediction or climate adaptation, the study was carried out within the context of precision agriculture with the goal of better crop disease and pest identification[44], [45].

Deep learning's potential uses in crop disease and pest classification, model improvement, software deployment, and validation were the primary focus of the research. It presented a systematic framework for incorporating Artificial Intelligence into agricultural decision support while taking into account constraints imposed by dataset size, context, and other technology factors. The results provided the groundwork for further studies to improve decision support systems in agriculture, increase the scalability of models, and enhance Artificial Intelligence driven agricultural solutions[46], [47].

1.7 Limitations of the Study

Problems with data collection, model creation, implementation, and validation were only a few of the study's limitations. When developing and testing the deep learning model, one of the main constraints was the accessibility and quality of the datasets used. The data was still lacking in geographic coverage, weather conditions, and crop types, despite attempts to use a varied dataset spanning crop diseases and pest infestations. Because of this limitation, the model could not be used universally; for example, it could not accurately classify some uncommon crop diseases or pests that are peculiar to a single area. It was also difficult to standardize the training data due to the prevalence of variations in image quality, illumination, and background noise produced by using publically accessible datasets and images obtained in the field[48], [49].

The computing demands of training and refining the deep learning model were another limitation. To reach its full potential, the hybrid CNN-SVM model needed a large amount of RAM, GPU acceleration, and computing power. The training process was held on a constrained hardware configuration due to resource restrictions, which made the experimentation process longer and prevented the model from being tested on bigger datasets. Time and technical limitations during model optimization and hyperparameter tuning may have also hindered the investigation of other configurations that may have yielded even better results. For contexts with restricted resources, such as small-scale farms with minimal computer power, real-time

processing becomes more difficult due to the model's complexity and the longer inference time it causes [50], [51].

Problems with accessibility and usability also affected the process of integrating the deep learning model into an intuitive software platform. Users could only upload images and get their classification results via the platform's web-based application design. The system's inability to function in remote locations with poor or unreliable internet service was a major drawback. The platform's lack of interactive feedback mechanisms meant that users couldn't manually fix misclassifications, which limited its adaptability and continuous learning potential.[52], [53].

Additional challenges were encountered while attempting to validate the generalizability and robustness of the model. There was insufficient real-world validation across various climates, soil types, and agricultural systems, but the model was evaluated on various datasets and compared to current deep-learning architectures. Additional insights into the practical issues of integrating AI-driven solutions into agricultural processes may have been gained via field-based implementation, however, the research mainly focused on controlled validation settings. Additionally, the model may not have been accurate in complex scenarios due to a lack of attention to external factors such as image occlusions, differences in disease development phases, and the existence of multiple crop diseases or pests in a single image[54], [55].

The research did not thoroughly investigate ethical and regulatory factors. Data privacy, bias in model predictions, and the possible implications of wrong classifications on farmers' decision-making were issues that arose with the incorporation of AI into agriculture. Legal regulations regulating AI-driven agricultural tools were not thoroughly addressed in the research. This is especially true concerning responsibility in instances where the model generated incorrect recommendations. The study improved decision-making tools for managing crop diseases and pests, but it missed a key opportunity to present a more complete image of the effect by ignoring the socio-economic reasons that drive farmers to choose AI-based solutions. Although there were several limitations, the study did a great work of laying the groundwork for future research on agricultural decision support systems powered by Artificial Intelligence[56], [57].

1.8 Contributions of the Thesis

The agriculture sector has several areas, including crop diseases prevention and pest control that stand to benefit greatly from the researcher's efforts in this thesis. First and foremost, creating and deploying a deep learning model optimized for decision support was a ground-breaking initiative that had the potential to completely alter the way in which key players in the crop health ecosystem think about their work. The model employed a state-of-the-art neural network architecture to shed light on the complex dynamics of crop disease and pest prevalence, giving key stakeholders a preventative resource for making well-informed decisions.

The thesis contributed significantly to the multidimensional subject of agricultural crop disease and pest control. The first primary objective was to identify the most important features of deep learning models that contribute to decision support. This insight added to the theoretical foundations of using deep learning in agriculture, expanding the body of information available for practical decision-making.

This study aimed to solve real-world problems by creating a state-of-the-art deep neural network model for detecting and classifying crop diseases and pests. The hybrid model of Convolution Neural Network and Support Vector Machine performed better than existing models, giving farmers and agricultural experts a more accurate and trustworthy resource. The development of an enhanced model represented a technical innovation that directly handled the complexity of crop diseases and pest identification.

The developed model was implemented in an agricultural setting to close the gap between theory and practice. By employing neural network architecture, the proposed model sought to offer timely and actionable decision support for farmers, hence contributing to the practical deployment of deep learning solutions in real-world agricultural settings. This was a necessary process for making the technology usable by real end users.

The model's evaluation across several crop disease types and pest variations added to the validity and reliability of the enhanced model. The goal of this validation was to determine how well the model generalizes across a wide range of conditions, as is experienced in practical agricultural settings. The final product of the thesis was a

well-rounded theoretical and practical contribution to agricultural decision-making in the area of crop disease and pest control.

1.9 Organization of the Thesis

To provide a thorough examination of the development of a deep learning model to assist with decision support in crop diseases and pest management, the thesis is methodically structured into eight chapters.

An overview of the research topic, objectives, justification, and importance of the study is provided in Chapter One, the Introduction. In it, the study's justification, scope, and constraints are laid out, along with the research questions.

In the second chapter, the researcher examined the current research on crop disease and pest classification using deep learning models. It discusses relevant publications, finds gaps in existing methods, and lays out the theoretical and conceptual basis for the research. Furthermore, it examines important deep-learning model components that influence decision support in crop diseases and pest management.

Research design, data collection methods, model development procedures, and performance assessment measures are all covered in depth in Chapter Three, which focuses on the research methodology. It lays down the steps used to develop the deep learning model, including acquiring the dataset, preparing it, and training it. In addition, the experimental setup is explained, including methods for hyperparameter tuning and validation, to guarantee the effectiveness and reliability of the model.

Factors influencing decision support in crop diseases and pest management that are particular to deep learning models are discussed in Chapter Four. Features selection, interpretability of models, data augmentation, and real-time processing capabilities are among the important components that are examined. This chapter's conclusions were the basis for the improvements made to the proposed model.

Improving the deep neural network model is the main subject of Chapter Five. For better classification results, it explains how to optimize the architecture, including Convolutional Neural Network (CNN) with Support Vector Machine (SVM), and more. To prove its effectiveness in detecting and classifying crop diseases and pests,

this chapter explains the model's training procedure, performance measures, and comparison with existing models.

The incorporation of the developed model into an intuitive software platform is discussed in Chapter Six. The chapter explains the model's incorporation into a web-based application that provides stakeholders with the ability to upload images, get classification results, and access treatment solutions that are proposed. To make sure that farmers, agricultural specialists, and decision makers can easily access and utilize the software, the chapter also explains its features, user interface design, and deployment method.

Verifying the model's robustness and generalizability is the primary emphasis of Chapter Seven. This ensures the model's trustworthiness in real-world agricultural settings by assessing its performance across various datasets with different crop diseases and pest varieties. Existing models, including DenseNet, AlexNet, EfficientNet, Visual Geometry Group, and Convolution Neural Network, are compared in the performance evaluation. Model scalability, flexibility, and possible biases influencing classification accuracy are also discussed in the chapter.

The summary, conclusion, and recommendations are all presented in Chapter Eight. It summarizes the main points of the study and shows how the research might help with decision-making when dealing with crop diseases and pests. The chapter concludes with several recommendations for how things may be improved in the future, such as expanding testing to other agricultural fields, integrating with agricultural monitoring systems that use the Internet of Things, and making models more explainable. At the end of the thesis, the researcher provides some recommendations for further study, stressing the need for ongoing model improvement and practical approaches to implementation.

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

This study's literature review looks into a wide-ranging investigation of the prior work in the fields of agriculture, Artificial Intelligence, and decision-support models. The review opens by stating the growing challenges faced by crop diseases and pests and the need for cutting-edge technology to combat them, underscoring the essential role that crop health plays in ensuring food security throughout the world [58].

To fully grasp the potential of deep learning in agriculture, it is essential to first familiarize oneself with the seminal work of Yong et al. [9]. They used convolutional neural networks in a novel way, creating a powerful framework for automating the diagnosis of diseases based on images. Later studies, such as Shubhajyoti et al. [6], used Recurrent Neural Networks (RNNs) for temporal modeling and predictive analysis, broadening the topic to encompass pest control. The use of deep learning in agriculture is becoming more multifaceted and multidisciplinary.

Moving on to decision support models, machine learning algorithms were included in decision matrices. The absence of a unified deep learning model that combines crop diseases and pests with decision support mechanisms for stakeholders stands out despite these advances as a significant gap in the literature. The goal of this thesis is to fill this knowledge gap by introducing a hybrid model that can detect and classify crop diseases and pests, while also providing decision support. It is hoped that contemporary agriculture would benefit greatly from this integration of deep learning technologies and decision-support tools [5], [59].

The in-depth literature analysis illuminates the complex state of knowledge at the crossroads of deep learning, crop health management, and decision support models. It provides background for the thesis, outlining the setting in which these components would be integrated and highlighting the possible influence on environmentally responsible farming and international food security[60], [61].

2.2 Theoretical Framework

This research was guided by theoretical frameworks, which offered perspectives that influenced problem identification, model selection, and results interpretation. This research used theories from the fields of information systems and artificial intelligence. Theoretical frameworks helped build a solid foundation for the deep learning model, which enabled the model to be used by the stakeholders in the real-world agricultural settings [19], [65]. The research was anchored on four main theories, all of which were in line with the research-specific objectives. The researcher identified and analyzed important factors influencing decision support systems based on the Decision Support Systems (DSS) Theory. An improved deep neural network model for the classification of crop diseases and pests was trained using the Classification Theory as its foundation. To guarantee usability and acceptance, the developed model was integrated into a user-friendly platform with the help of the Technology Acceptance Model (TAM). Finally, the model's generalizability and applicability across multiple agricultural conditions were validated using the Generalizability Theory. Bringing together theories from validation, decision science, machine learning, and human-computer interaction, these theories created a multidimensional foundation. They made sure that the research contributed both to theory and practice [20], [66].

2.2.1 Decision Support Systems Theory

First, this research aimed to identify and investigate important factors influencing decision support in crop diseases and pest management. The Decision Support Systems (DSS) Theory offered a good platform for this purpose. The Decision Support Systems theory, which has its origins in decision science more generally, suggests that computers should complement human decision-makers rather than replace their role. Decision Support Systems are supposed to improve decision quality, speed, and efficacy in complex and unpredictable contexts by integrating data, analytical models, and user-friendly interfaces.

In the past, Decision Support Systems theory developed as a result of the realization that managers and stakeholders often encountered difficulties while addressing semi-structured or unstructured issues that could not be handled with standard information

systems alone [20]. These difficulties were most noticeable in agricultural settings when it came to managing crop diseases and pests, as decisions were impacted by a number of factors, including crop type, model architecture, data quality, and available treatment methods. In order to enable evidence-based agricultural decisions, the Decision Support Systems framework therefore emphasized the need to combine real-time data, classification models, and expert knowledge.

Key factors that shaped effective decision support for stakeholders were identified and analyzed in this research using Decision Support Systems theory as a framework. The approach emphasized that decision support was multidimensional, depending on factors such as the usability of the system, the context of deployment, and the correctness of data and models. It underlined that deep learning models needed to provide practical insights that met the demands of agricultural stakeholders like farmers and extension agents in order to have an impact.

The notion that decisions in agriculture are often dynamic and need continuous learning and adaptation was further supported by Decision Support Systems theory [8]. Based on this theory, the study's first objective found that technical performance alone wasn't enough to evaluate a deep learning model's efficacy in managing crop diseases and pests; other important metrics included the model's capacity to integrate multiple data sources, provide timely decision-support, and boost stakeholder confidence in managing agricultural risks. Thus, the research advanced technological innovation while addressing the real-world complexities of decision-making in agriculture, which was anchored on the Decision Support Systems theoretical foundation [30], [67].

2.2.2 Classification Theory

The second objective of this research was to develop an enhanced deep neural network that could detect and classify crop diseases and pests. The Classification Theory was the theoretical basis for this purpose. Based on their common features and attributes, data or objects are systematically grouped into predetermined classes according to classification theory [38]. Finding discriminative patterns in the data that allow one class to be differentiated from another was used as the basis for effective classification.

Statistical learning and pattern recognition are the historical theories that classification theory evolved from, which aim to minimize misclassification errors by developing methods to assign data points to classes. Decision Trees, Support Vector Machines, and Neural Networks are computational models that learns from data to enhance classification accuracy, and they are a natural extension of classification theory with the advent of machine learning and Artificial Intelligence. Classification played a crucial role in agriculture by allowing for the detection of small variations in color, texture, and form within crop images, which were used to differentiate between healthy crops and those affected by diseases and pests.

To make sure the model could correctly extract and use significant features for classification, classification theory guided this research in the development of the deep learning architecture. In addition to ensuring technical correctness, the theory emphasized that classification should provide meaningful classes that should guide stakeholders' decision-making. For instance, in order to recommend treatments like specialized fungicides or pesticides, it was crucial to differentiate between various forms of maize leaf diseases or pest infestations.

In addition, validation was crucial for verifying the model's generalizability across different datasets, as classification theory highlighted. To avoid overfitting and keep up good classification performance in real-world conditions, the theory emphasized that a classification model should be able to generalize to new data. The research acknowledged that the hybrid CNN-SVM model's capacity to reliably and accurately classify crop diseases and pest infestations was vital to its success, and this is why the second objective was grounded on the classification theory. Thus, the classification theory made sure that the study contributed to an enhanced deep learning model while also providing useful information for stakeholders to use in decision-making.

2.2.3 Technology Acceptance Model

The third objective of this research was to integrate the developed deep learning model into an easy-to-use platform, where stakeholder acceptance and usability could be guaranteed by using the theoretical foundation provided by the Technology Acceptance Model (TAM). The original Technology Acceptance Model was developed by Davis in 2015. It used two main components, Perceived Usefulness (PU) and Perceived Ease of Use (PEOU), to explain and predict how people would embrace new technology [55]. The theory behind the concept was that people's opinions about a system affected their attitude towards utilizing it, which in turn dictated whether or not they would actually utilize the technology [62].

In the field of information systems research, the Technology Acceptance Model has become a leading theory because of its simplicity and resilience throughout the years [43]. By including new variables, like social influence, trust, and enabling circumstances, the model was able to be extended and used in a wider range of contexts [45]. Using the Technology Acceptance Model, researchers in the agricultural sector were able to deduce that perceived advantages and ease of use were crucial factors in the acceptance of decision support systems, precision farming technology, and mobile applications [67].

Both the technical accuracy and the acceptability of the deep learning model by stakeholders like farmers and extension officers were crucial to its performance in this research, making the Technology Acceptance Model relevant. According to the hypothesis, even cutting-edge technology might fail if people find it too complicated or difficult to use [48]. The study acknowledged that stakeholder adoption was influenced by both functional and experience variables, which is why the third objective was grounded in the Technology Acceptance Model. Stakeholders were more motivated to include the platform in their day-to-day decision-making if, for instance, it was seen as user-friendly and made practical recommendations like pesticide choices and agrovet locations.

To further enhance system usability and guarantee conformity with stakeholder expectations, the Technology Acceptance Model highlighted the significance of iterative testing and user input [49]. This viewpoint informed the user interface design

and feature integration, turning the developed model into a useful decision support tool instead of just a technological innovation. As a result, the Technology Acceptance Model reaffirmed the notion that developing an accurate deep learning model was only one metric for research success; the acceptance, usability, and continued usage of the model in actual agricultural contexts were equally important.

2.2.4 Generalizability Theory

Validating the robustness and adaptability of the developed model across varied agricultural contexts was the focus of the fourth objective of this research, which was theoretically supported by the Generalizability Theory. In 2016, Cronbach and other authors came up with Generalizability Theory, which expanded traditional test theory and addressed the shortcomings of reliability estimates that were based on a single source of measurement error [71]. It proposed that several aspects of measurement—including subjects, items, raters, and contexts influenced observed scores, and that separating these sources enabled a more precise assessment of reliability and generalizability [72].

Generalizability Theory offered a framework for identifying and measuring various types of measurement error, resulting in more comprehensive reliability estimates [63]. This is in contrast to conventional test theory, which only produced a single reliability coefficient. The reliability of measures under varied conditions was evaluated using this theoretical method, which has found extensive use in the fields of education, psychology, and health sciences [45]. Models need to be tested for robustness across varying datasets, settings, and contexts; hence, it was also adopted in computational and machine learning domains in recent years [66].

Generalizability Theory played an important role in this research as the deep learning model was trained to identify and classify various crop diseases and pests in a wide range of real-world scenarios. Theoretically, it was necessary to evaluate model performance over a variety of agricultural contexts, crop types, image quality standards, and pest variances in order to draw valid conclusions about its efficacy [7]. The research used Generalizability Theory concepts to acknowledge that other factors, including illumination, background noise, or geographical variances, affect classification results.

In order to measure the reliability of the developed hybrid CNN-SVM model, Generalizability Theory emphasized the significance of doing cross-validation and comparative analysis with other models [58]. This theoretical foundation made sure that consistency, fairness, and adaptation across various contexts were also considered while evaluating robustness, in addition to accuracy. The research assured that the developed model could be used in different agricultural contexts and not limited to controlled experimental circumstances by grounding the fourth objective in Generalizability Theory, thereby addressing the key problem of external validity. By presenting validation as a multi-dimensional process, Generalizability Theory improved the research's methodological process, making the model more trustworthy and suitable for supporting agricultural decision-making.

The theoretical framework lays the groundwork for an all-encompassing strategy for building a deep learning model to aid in pest and crop disease management decisions. The framework's goal was to build a flexible system that helps farmers practice sustainable and resilient agriculture by incorporating various data sources, using advanced model architectures, putting transparent decision support mechanisms in place, and involving stakeholders at every stage [68], [69].

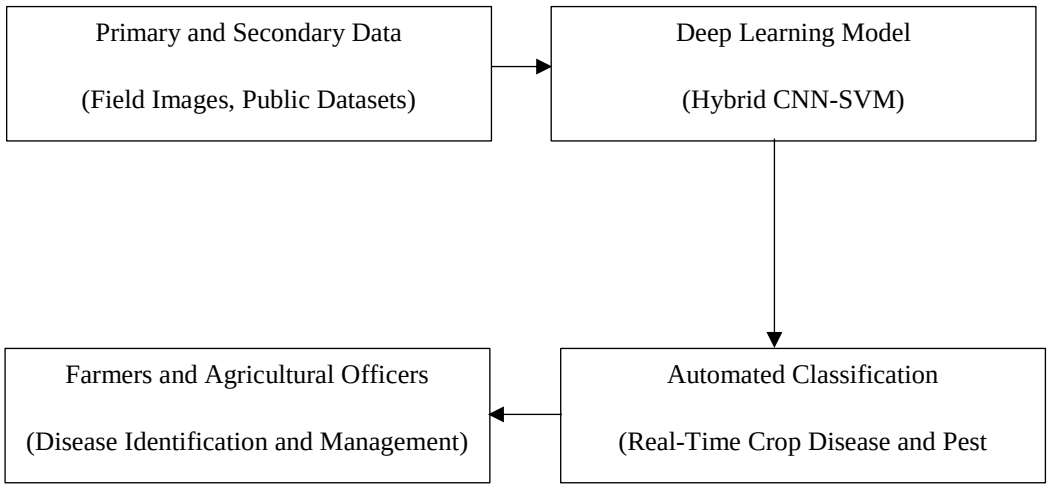


Figure 2.1 Theoretical Framework for Deep Learning Based Decision Support

The datasets and external variables that were crucial to building the deep learning model were included in the Information Sources component. Field images taken by mobile devices were among the major data sources, while secondary data came from publicly accessible databases. By including these data sources, the researcher made

sure the model was trained on a large and representative dataset, which improved its predictive power and resilience [70], [71].

To classify crop diseases and pests, the model design phase optimized and developed a deep learning architecture. To increase classification accuracy, a model that combines CNN with SVM was developed. This model made use of CNN's feature extraction capabilities and SVM's robust classification performance. To make the input data better, image preprocessing methods were used, such as removing noise, adding more data, and extracting features. To make sure the model could adapt to different agricultural conditions, the researcher used hyperparameter tuning and optimization techniques, including transfer learning and fine-tuning, to make it work better [72], [73].

A key emphasis of the decision-making tool provides stakeholders with practical insights derived from the model's predictions. To intervene quickly in the event of crop diseases or pest infestations, the automated classification system enabled real-time identification. To help users find the best insecticides, fungicides, and agricultural methods, the application had a recommendation system. To further assist farmers and agricultural authorities in locating nearby agrovet outlets that provide pest and disease control products, a geolocation service was also included. To successfully reduce the impact of pests and diseases on crops, this decision-making tool made sure that users got accurate and timely information [74], [75].

For the deep learning model to be adopted and implemented successfully, the stakeholder participation phase was vital. The method helped farmers and agricultural officials identify crop diseases and prescribe remedies, which increased harvest yields. By increasing the dataset size, enhancing the classification accuracy, and adding new characteristics, researchers and data scientists helped develop the model. Government officials and agricultural groups used the system's findings to guide policymaking, improve agricultural extension services, and create methods for managing pests and diseases on a grand scale. Maintaining the model's relevance, use, and adaptability to real-world agricultural challenges was achieved via the collaborative engagement of stakeholders [76], [77].

2.3 Factors that Influence Decision Support in Crop Diseases and Pest Management

Incorporating deep learning models into decision support systems for managing crop diseases and pests is a game-changing strategy for resolving problems in the agriculture industry. To better assist stakeholders in crop health management with their decision-making, this research delves deeply into the aspects that affect the efficacy and usefulness of deep learning models. There are several factors that contribute to the success of these decision support solutions, including data quality and model design, real-time updates, interpretability, user-friendliness, ethical issues, and scalability[10], [78].

i. Diversity and quality of data

The strength of a deep learning model relies on the quantity and variety of its training data. The model relies on a large dataset that includes information on past disease patterns, environmental factors, and pest prevalence across a wide range of crops and geographic areas in order to spot trends and generate reliable forecasts. For the sake of training a model that can generalize effectively to other agricultural conditions, it is crucial to ensure that the dataset is diverse and representative [11], [79].

ii. Model Architecture

The effectiveness of a deep learning model for controlling crop diseases and pests hinges on the architecture chosen for the model. The chosen architecture must coincide with the characteristics of the input data and the particular needs of the decision support system, whether it be a Convolutional Neural Network (CNN) for image-based crop disease identification or a Recurrent Neural Network (RNN) for temporal modeling. Model complexity and interpretability must be balanced for effective implementation and comprehension by end users [12], [80].

iii. Real-time Data Updates

As a rapidly evolving industry, agriculture relies heavily on accurate and timely data. Crop health and pest incidence may vary quickly, needing decision support algorithms to adjust in real-time. By including methods for continual data updates, the model can adapt to changing agricultural circumstances and provide stakeholders with accurate and timely data for decision-making [11], [81].

iv. Interpretability and Explainability

To gain the trust of skeptical stakeholders, the results of a deep learning model must be clear and explainable. To effectively manage crop diseases and pests, farmers, agricultural specialists, and policymakers must have a firm grasp of how the model arrives at its precise recommendations. Stakeholder trust in the decision-making process may be bolstered, and openness can be increased with an easily interpretable model [12], [82].

v. User-Friendly Interfaces

Interfaces that are easy to use and accessible have a significant role in determining whether or not deep learning models are put to use in agriculture. The success of the decision support system depends on the ease with which stakeholders are able to understand and implement the system's recommendations. Accessibility and usability are enhanced in real-world agricultural contexts via user-friendly interfaces [13], [83].

vi. Integration with Existing Agricultural Practices

The successful use of deep learning models requires that they be seamlessly integrated with current agricultural practices and decision-making processes. The adoption and use of the model will be facilitated by matching its results to existing procedures and decision criteria. Agricultural practices may be improved with the help of a well-integrated model [14], [84].

vii. Ethical and Responsible Artificial Intelligence

Ethical issues must be included in the process of creating and releasing deep learning models. Ethical Artificial Intelligence in agriculture requires efforts to be made to reduce bias, ensure equality, and gather and use data in a responsible manner. Responsible Artificial Intelligence is in accordance with the ideals of justice and accountability, and it includes practices like regularly inspecting the model for biases and adherence to ethical rules [15], [85].

viii. Scalability and Adaptability

Deep learning models must be scalable and flexible to be widely used across a wide variety of agricultural settings. Models are more likely to be useful and widely adopted if they are designed to be applied to a wide range of crops, geographic locations, and

agricultural systems. A model's extensive adoption and influence depend on its flexibility to accommodate a wide range of circumstances and user demands [16], [86].

A nuanced assessment of numerous parameters is necessary for the effective adoption of deep learning models for decision support in crop diseases and pest control. There are several factors that affect how effective and feasible these new approaches are. These include data quality and model design, as well as real-time updates, interpretability, user-friendliness, ethical issues, and scalability. Decision support systems in agriculture might be greatly improved by deep learning models if they were to take into account all of these elements. This would help to spread environmentally friendly farming methods and increase worldwide food production [87], [88].

Table 2.1 Deep Learning Model Factors

S/N	Factor	Description
1	Data Quality	Accurate, reliable, and well-labelled data improves model training and prediction accuracy.
2	Data Quantity	Larger datasets improve model generalisation and performance.
3	Feature Representation	Effective features such as images, sensors, and weather data improve disease and pest identification.
4	Model Architecture	Selecting suitable deep learning architectures (e.g., CNNs) enhances pattern recognition and classification.
5	Transfer Learning	Pre-trained models improve performance when labelled agricultural data is limited.
6	Hyperparameter Tuning	Optimising parameters such as learning rate and batch size improves model efficiency.
7	Explainability	Interpretable models increase trust and adoption by explaining prediction decisions.
8	Real-time Capability	Fast predictions support timely disease detection and pest management decisions.
9	System Integration	Integration with farm systems, IoT, and remote sensing enables comprehensive crop monitoring.
10	Continuous Updating	Regular model updates maintain accuracy as disease and pest patterns change.
11	Ethical Considerations	Ensures responsible use through privacy, fairness, and accessibility considerations.

2.4 Explainability and Trust

Researchers sometimes refer to deep learning models as black boxes because of their complicated structures and large number of parameters; this makes it difficult for users

to trust and accept these models. To make sure that people like farmers and agronomists can understand and believe the predictions and suggestions made by a deep learning model for decision support in pest and crop disease management, the model must be trustworthy and easily explainable [8], [89].

Many different strategies have been used in the field's research to address these difficulties. A popular approach is to use Explainable Artificial Intelligence (XAI) approaches, which seek to shed light on how deep learning models function. In order to understand how certain inputs affect a model's output, methods such as Shapley Additive Explanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) is used. Transparency and trust are enhanced by these technologies, which let stakeholders see and understand the characteristics and aspects that go into making a certain prediction [9], [90].

Making deep learning models that are naturally interpretable is another strategy. A few examples of this kind of design include simpler ones, hybrids of neural networks and decision trees, or rule-based systems, or attention mechanisms that draw focus to the most important aspects of the incoming data. Users are able to better understand the model's reasoning using these designs, which in turn increases their trust in its results [10], [91].

Furthermore, establishing trust requires contextualization. Finding ways to explain things that are consistent with the expertise of those who have a stake in the outcome is a common theme in studies conducted in this area. Making use of visual aids such as heatmaps or annotated photos, explaining things in simple terms, and presenting recommendations that stakeholders may verify with their knowledge are all part of this [11], [92].

Deep learning models that are both effective and easy to comprehend are the goal of current research, which combines several methodologies in the hopes of achieving these goals. More informed and effective decision-making in agricultural operations is made possible by these techniques, which also boost the desire of stakeholders to utilize the technology for a decision support system in crop diseases and pest control [12], [93].

2.5 Deep Neural Network Model Development Approaches

Mohanty et al., using a public dataset collected and under controlled conditions, analyzed 54, 306 healthy and diseased images. The 14 crop species were analyzed, and a total of 26 diseases were classified from all images in the dataset. The deep learning approach used indicated that transfer learning using GoogLeNet with the plant village dataset yielded an image classification accuracy of 96.34%, while training from scratch using AlexNet resulted in an accuracy score of 85.53%. The models were able to generalise the training data well since the techniques were set to use 30 epochs until the convergence point was reached. An accuracy of 96.21% was still produced by the transfer learning method when just 20% of the training data and 80% of the test data were used. The overfitting problem did not affect the classification accuracy that much, since increasing the test set to train set ratio, the accuracy did not drop drastically[10], [93].

In their 2023 publication, Vinoth et al. [11] described a method for classifying pneumonia diseases using the AlexNet deep learning algorithm. The respiratory system is compromised by the deadly disease known as pneumonia. Despite the widespread belief that viruses are the most common cause of pneumonia, radiation is still a common tool in the treatment toolbox. Examining chest X-rays is necessary for the diagnosis of pneumonia. The main problem for countries attempting to reduce the transmission of pneumonia is the lack of a portable diagnostic instrument. Proof that researchers consider automated diagnostic systems valuable and necessary. Research conducted by the World Health Organization (WHO) revealed that a radiologist is still required to assess two-thirds of pneumonia patients. Here, the AlexNet model was proposed as a solution to these problems. The proposed approach primarily aimed to diagnose pneumonia early on by using the AlexNet model. The researchers utilized Kaggle's dataset, which contained both training and testing data, to put the solution into action. The tests evaluated the accuracy and efficiency of the AlexNet model using a variety of performance metrics. With a 96% success rate, 93% accuracy, and 98% recall, the data clearly shows that the proposed Alexnet model was accurate[94], [95].

Using an experiment including ANN and Support Vector Machine, Pujai et al. [9] categorized images of plant diseases. A feature extraction approach was used to train

the two classifiers utilizing image features. When testing with the retrieved image features, the findings showed that SVM outperformed ANN. The ANN classifier achieved an accuracy of 0.874, while the SVM classifier achieved a precision of 0.9217 [96].

Using machine learning algorithms, Yaktundimath et al. [10] were able to identify and categorize three species of cereal plants. Using the specific fungal symptoms linked with each leaf disease, they trained two classifiers to distinguish between jowar, maize, and wheat leaf diseases. The scientists collected healthy, smut, powdery mildew, leaf spot, and leaf blight-infected maize leaves for the experiment. The scientists obtained 750 JPG-formatted images, some of which were healthy and some of which were fungal-affected, and then followed specific protocols to detect and classify indications of fungal diseases. Next, the images underwent preprocessing before the segmentation process began. The Color Co-occurrence matrix approach was used to extract features from images of leaf diseases. These features were then passed into machine learning algorithms. The program interface was implemented using the MATLAB tool. The classification accuracy for the SVM algorithm was 83.83%, while the ANN algorithm achieved 77.75%. After comparing SVM and Artificial Neural Network algorithms, the authors concluded that SVM is superior at identifying and classifying cereal fungal disease [10]. The study concluded that more work is required to categorize leaf disease images using machine learning techniques and other feature extraction approaches [97].

Five distinct crop diseases were classified by Zhang et al. [11] using machine learning techniques. To experiment, they gathered 20 images for each type of leaf disease; these images served as both training and testing sets. The five-leaf diseases that were tested in the experiment, however, were not mentioned by the authors. After the images were adjusted for size and orientation, histogram equilibrium was used to make them 32 by 32 pixels. Each pixel in the images had a 255 gray level, thus they all had a white backdrop. The experiment began with the collection of digital camera images of five distinct leaf diseases, followed by the segmentation of these images. Afterward, the KNN algorithm was used to extract features from the images. The features were then categorized according to the specific leaf diseases, resulting in class labels for each feature in the images. Fifty scores of the image classification experiment yielded

results showing a categorization accuracy of more than 80%. Lastly, the researcher suggested expanding the training data set and focusing on image key point extraction for future study, since these points serve as distinct image differentiators [98].

Research conducted in the China farm region by Xiaoyang et al. [12] identified four distinct plant leaf diseases; the researchers employed digital cameras to capture JPG images of maize diseases in natural light. After converting the images to BMP format, they used a thresholding value to separate the images and extract the data. After the images were transformed from RGB to HIS and the GA-SVM algorithm was used for image classification, the researchers computed the standard deviation and mean. The classification of diseases affecting maize leaves also made use of the Support Vector Machine and the RBF kernel function. The accuracy with which the machine learning algorithm identified the diseases affecting maize leaves was evaluated using precision. In comparison to the Support Vector Machine's accuracy of 69.63% to 90.09%, the GA-SVM algorithm's accuracy ranged from 88.72% to 92.59% across all images [99].

Using the Support Vector Machine technique for image classification in combination with the HOG and LBP feature extraction methods, Mohammad, Sayeed, and Billah developed a model for plant disease diagnosis in 2019. They used the Flavia leaf dataset, which is accessible to the public, for both training and testing purposes in their investigation. The researchers used a Support Vector Machine for image classification and the Local Binary Pattern and Histogram of Oriented Gradients for feature extraction. By combining the Local Binary Pattern with the Histogram of Oriented Gradients, a Support Vector Machine was able to classify the extracted image features with a classification accuracy of 0.9125. The accuracy of 0.406 obtained by feeding features extracted using the Local Binary Pattern into the Support Vector Machine classifier showed that the Local Binary Pattern feature extraction approach was not very effective. Finally, the comparable accuracy values were 0.8531, 0.8125, and 0.775 when a Support Vector Machine was used as a classifier, and the Histogram of Oriented Gradients was used as a feature extraction method. The image sizes utilized were 8 by 8, 4 by 4, and 2 by 2, respectively. The greatest results were achieved when using an 8x8 image cell size in conjunction with the Support Vector Machine classifier and the Histogram of Oriented Gradients feature extraction method. The experimental results showed that the best feature extraction from images using a Support Vector

Machine Classifier was achieved by combining Local Binary Pattern and Histogram of Oriented Gradients [13], [100].

By using machine learning techniques, Panigrahi et al. [14] were able to detect and classify plant leaf diseases. A classification accuracy of 79.23% was attained using the supervised machine learning technique known as Random Forest. In order to conduct the experiments, the dataset, which included 3823 images, was obtained from the Plant Village website. The images were categorized as follows: common rust (1,192 images), gray leaf spot (513 images), northern leaf blight (956 images), and healthy (1,162 images). The researchers used these images to train and test the image classification algorithm. Because processing enormous images often requires a lot of memory and effort, image processing often involves scaling down the images to a consistent size. Because processing features from grayscale images is simpler and quicker than from colored images. Image segmentation, which included dividing the images into smaller parts and removing unnecessary ones, was also a part of the experiment. The researchers used the most important and distinguishing parts of the images to extract features since that's where all the important and unique details were located. The features were extracted and classified according to shape, colour, and texture since the models were very effective at identifying and classifying images of maize leaf disease using these features. Keep in mind that the 3823-image maize disease dataset served as the basis for feature extraction. A 90% training set and a 10% test set were used to train machine learning algorithms on the divided dataset. Windows 7 with Python 3.3 installed was used to set up and implement the SVM, KNN, RF, DT, and NB classifiers. Python software was used in conjunction with the Pandas package and the Python machine learning library. In order to ensure that all of the images in the dataset were the same size, they were downsized to 100 by 100 pixels. To transform grayscale images into a CV2 format, the Python image library was used, which is called CV. The prepared images were fed into a pickle, which performed the process again. After processing was finished, the pickle file was fed into the classification models [101]. The data used to train the models determines the diseases that will be predicted. To choose the best model on the test data set, F1-score, recall, accuracy, and precision were used to assess each model. To distinguish between healthy and maize leaf diseases, and to identify which disease has affected each leaf,

the models used a classification system for maize leaf diseases. By doing so, farmers were able to identify the most effective fungicide to treat maize leaf disease and take precautions to avoid additional infection of the leaves [14].

A model for crop leaf recognition was developed by Zixi et al. [15] using principal component analysis and a Support Vector Machine. Experiment subjects included maize leaves infected with corn rust, northern leaf blight, healthy leaves, and grey leaf spot. For background segmentation in images, OpenCV morphological transformation and morphological operation methods were used to create an object outline and a segmentation mask. The experiment was conducted to determine the difference between the features of the maize leaf diseases. Then, a complete image of the maize leaf was obtained, which was then processed using a Support Vector Machine and Principal Component Analysis. Using a linear support vector machine kernel with a 100-value penalty parameter, the classification accuracy for four different diseases was 0.9550,0.9264,0.9123, and 0.9578, respectively. The image dimensions were reduced using Principal Component Analysis (PCA) to save storage space. Its primary goal was to generate a new feature vector to feed into the support vector classification model by calculating k-dimensional features from the n-dimensional data. Classification of the processed dataset using the support vector machine yielded better results [102], [103].

To check whether the images showed diseased leaves, Chad DeChant et al. [104] developed a three-step analytic process. Through the use of the Bisque platform on CyVerse, a grand total of 1796 images were acquired for the purpose of data gathering; out of these, 1028 showed infected maize leaves, and 768 showed healthy leaves. In order to assess whether or not the whole collection of images included infected plants, a Convolutional Neural Network (CNN) was trained using the collected images. This model's performance was optimized using a three-stage design. The model performed well, with an impressive 96.7% accuracy rate. Deep learning was used by Xihai Zhang et al.[105] to propose the GoogLeNet and Cifar10 models for diagnosing leaf diseases. About 500 images of maize leaves infected at different periods and places throughout history were compiled from various sources such as Google and Plant Village. Nine distinct groups were then created from these images. With an average recognition accuracy of 96.1 percent, the GoogLeNet model classified eight distinct diseases

affecting maize leaves. The results demonstrated that various modifications to the model's parameters, as well as the inclusion of a Relu function and dropout technique, significantly improved classification accuracy [106], [107].

Using a smartphone camera, images of maize leaf diseases from several maize fields, and the Plant Village online website, Malusi Sibiya et al. [108] trained a Convolution Neural Network to recognize and categorize the diseases. Grey leaf spot (Cercospora), northern corn leaf blight (Exserohilum), and common rust (Puccinia sorghi) were selected as the three maize leaf diseases to study. The CNN's accuracy in detecting and labeling northern maize leaf blight (Exserohilum) was 99.9 percent. An improved DenseNet architecture was proposed by Abdul Waheed et al. [109] for disease detection and classification in maize leaves. In addition, four more popular CNN models, EfficientNet, XceptionNet, NASNet, and VGGNet, were trained for the purpose of identifying and categorizing maize leaf diseases. The grey leaf spot caused by Cercospora (1644 images), common rust (3816 images), healthy crop (3720 images), and northern leaf blight (3152 images) were only four of the categories into which a total of 12,332 images of crops were organized. The proposed DenseNet yielded remarkable results, with an accuracy of 96.06% [110], [111].

Plant disease detection using deep learning was suggested by Md. Ashraful Haque et al. [34]. It was proposed that deep learning be used to detect images of maize leaf diseases in the field. The images were taken in Ludhiana, India, on the ICAR-IIMR's experimental grounds. The experimental areas in the three maize growing zones were photographed, yielding a total of 5,939 images of the crop in the field. Three major diseases, including Turcicum Leaf Blight, Maydis Leaf Blight, sheath Blight, and Banded Leaf, were photographed alongside healthy leaves using digital cameras and cellphones. The maize dataset was used to simulate three different types of network topologies. The testing results showed that on a separate test dataset, Inception-v3 GAP achieved a maximum accuracy of 95.99% [112], [113].

Zhang et al. [4] used a GA-SVM classifier model with rotationally orthogonal parameters in their research. After making the appropriate parameter choices, it was determined that the optimal values for population size (M), crossover frequency (Pc), and mutation rate (Pm) were 50, 0.7, and 0.05, respectively, yielding a 92.82%

accuracy in classification. Six distinct transfer learning models, including VGG-16, VGG-19, VGG-20, ResNet50, ResNet101, and ResNet152, were utilized in another work published by Liu et al. [11] (2020). Additionally, the authors tried out accelerated R-CNN models using a variety of image sizes. Both desktop computers and mobile phones were used to capture the images used in the experiment. For the UAV data, the ResNet101 model showed a maximum accuracy of 94.99%. Machine learning has not only enhanced the agricultural domain, but also the financial sector [12], the industrial sector [13, 14], and the crop disease detection system [15], [114].

The 2019 model for plant disease identification was developed by Muhammad, Sayeed, and Billah using the Support Vector Machine image classification technique in conjunction with the HOG and LBP feature extraction techniques. In their study, they used a publicly available dataset, the Flavia leaf dataset, for both training and testing purposes. Before using a Support Vector Machine for image classification, features were extracted from the images using the Local Binary Pattern and Histogram of Oriented Gradients. An accuracy of 0.9125 was achieved when the Support Vector Machine was used for classification in conjunction with Local Binary Pattern and Histogram of Oriented Gradients for feature extraction [115]. A low accuracy of 0.406 was achieved using the Local Binary Pattern feature extraction approach in conjunction with the Support Vector Machine classifier, suggesting that the approach did not produce good results. Finally, using a Support Vector Machine classifier and a Histogram of Oriented Gradients as a feature extraction approach yielded accuracies of 0.8531, 0.8125, and 0.775, with a cell size of 8, 4, and 2 pixels, respectively. The Support Vector Machine classifier produced the best results when applied to 8x8 images using the Histogram of Oriented Gradients feature extraction method. Hence, the Support Vector Machine Classifier [13] achieved the best results for image classification while using a hybrid of Local Binary Pattern and Histogram of Oriented Gradients as a feature extractor [116], [117].

To check whether the maize leaf disease images showed diseased leaves, Chad DeChant et al.[5] developed a three-step analytic process. The data collection included 1796 images of maize leaves, 1028 of which were NLB-infected and 768 of which were uninfected; the images were gathered using the Bisque platform of CyVerse. A fully Convolution Neural Network (CNN) was trained to assess whether or not the

whole collection of images included infected plants based on the predictions that matched. This model's performance was optimized using a three-stage design. The model performed with an impressive 96.7% accuracy rate. To diagnose leaf diseases, Xihai Zhang et al. [6] proposed the deep learning-based GoogLeNet and Cifar10 models. About 500 images of maize leaves infected at different periods and places throughout history were compiled from various sources such as Google and Plant Village. These images were then classified into one of nine groups. The GoogLeNet model accurately identified eight distinct maize leaf diseases with an average accuracy of 96.1%. The results demonstrated that multiple model parameter tweaks, as well as a greater variety of pooling operations, a Relu function, and dropout procedures, all improve classification accuracy [118], [119].

To detect and classify maize leaf diseases, Malusi Sibiya et al.[7] utilised a smartphone camera to collect images of maize leaf diseases from several maize farms, and the Plant Village online website to train a CNN network using Neuroph. The maize leaf diseases selected were grey leaf spot (Cercospora), northern corn leaf blight (Exserohilum), and common rust (Puccinia sorghi). When it came to identifying and categorising northern maize leaf blight (Exserohilum), the CNN had a high accuracy of 96.9%. An improved DenseNet architecture was proposed by Abdul Waheed et al. [8] for disease detection and classification in maize leaves. For the detection and classification of maize leaf disease, they also trained four more popular CNN models, including EfficientNet, XceptionNet, NASNet, and VGGNet. Careful curation of 12,332 images from diverse sources resulted in four distinct categories of crops: grey leaf spot caused by Cercospora (1644 images), common rust (3816 images), healthy crop (3720 images), and northern leaf blight (3152 images). The proposed DenseNet yielded remarkable results, with an accuracy of 97.06%.

Md. Ashraful Haque et al.[9] proposed using deep learning algorithms to detect crop diseases. It was proposed that deep learning would be used to detect images of affected crops in the field. The images were taken in Ludhiana, India, on the ICAR-IIMR's experimental grounds. The crop images taken in the field were 5,939 across three different experimental regions. The Turcicum Leaf Blight, Maydis Leaf Blight, sheath Blight, and Banded Leaf were photographed beside healthy leaves and against different backdrops using digital cameras and cellphones. Three different network

topologies were simulated using the maize dataset. The testing results showed that on a separate test dataset, Inception-v3 GAP achieved a maximum accuracy of 95.99% [120], [121].

Priyadharshini et al. [16] (2019) provided an alternative study for plant leaf disease identification, in which they offered an updated version of the LeNet model, also based on the Convolutional Neural Network. The model was trained using a gradient descent approach. In 2021, Sujatha et al. [17] presented a comparison study. The research compared the efficacy of deep learning and machine learning-based models for identifying diseases in plant leaves. Stochastic Gradient Descent (SGD), Random Forest (RF), Support Vector Machine (SVM), Inception-v3, VGG-16, and VGG-19 were among the models that experiment participants selected. The VGG-16 model performed the best, whereas the Random Forest model had the worst results. In 2020, Panigrahi et al. [18] introduced a Convolutional Neural Network (CNN)-based method for crop disease identification. Adam was used to find the best possible parameters for the model. In terms of precision, the model did marvelously. Aravind et al. [19] (2018) also presented research to classify diseases affecting maize. The authors developed a multiclass SVM strategy using a bag of features [122], [123].

Mohanty et al. [17] analyzed 54,306 images of healthy and crop diseases using a publicly available dataset gathered in a lab. All images in the dataset were analyzed for each of the 14 crop species, and 26 diseases were identified. While training from scratch with Alexnet resulted in an accuracy score of 85.53%, the deep learning technique used showed that transfer learning using Googlenet with the PlantVillage dataset resulted in an image classification accuracy of 94.34%. Models could generalize the training data successfully since the models were set up to use 30 epochs until the convergence point was reached. When 20% of the training data was used, and 80% of the test data was used, the transfer learning technique still achieved an accuracy of 96.21%. Since the accuracy of classification did not substantially decrease when the test set to train set ratio was increased, overfitting did not seem to be a major issue [124], [125].

Using machine learning techniques, Yaktundimath et al. [10] were able to distinguish between three different types of grain plants. In particular, they trained two classifiers

to identify crop diseases using fungal symptoms associated with jowar, maize, and wheat leaves. To experiment, the researchers collected samples of healthy, powdery mildew, leaf spot, and leaf blight maize leaf diseases. The researchers used a systematic approach to recognize and classify fungal disease symptoms; they collected 750 healthy and fungal-infected images in JPG format. After that, image preprocessing and segmentation were carried out. Features of maize leaf diseases were extracted from images using the Colour Co-occurrence matrix method, with the MATLAB tool serving as the software interface. These features were then fed into machine learning algorithms. The classification accuracy for the SVM and ANN machine learning techniques was 83.83% and 77.75%, respectively. In terms of identifying and classifying cereal fungal disease, the researchers found that the SVM algorithm outperformed the Artificial Neural Network[10]. The research highlighted the need for more research into other ways of extracting features and machine learning techniques for classifying images of maize leaf diseases [126], [127].

Image classification using a deep learning model requires training the model on the images. One of the most important requirements is diagnosis, and there are several things that need to be looked at. This led to the development of a model that uses deep Convolutional Neural Networks to help doctors make better decisions. In the experiment conducted, the MobileNet model produced better results for the same dataset compared to machine learning algorithms[20]. The study concluded with an analysis of algorithms running on various mobile platforms. When compared to Google Net, MobileNet's accuracy proved to be better [21]. Worldwide, cancer is a huge public health concern. The most prevalent malignancy among females, breast cancer, develops when aberrant cells proliferate uncontrollably. There are significant obstacles to detecting and classifying breast cancer [22]. Hence, many state-of-the-art computational methods have been used in breast cancer detection and classification. Such models include Decision Tree, Evolutionary Algorithms, K-Nearest Neighbour (KNN), and Support Vector Machine (SVM)[23], [128].

Nonetheless, there is a wide range of possible levels of accuracy when using each model. The researchers also proposed a novel CNN model that is built on VGGNet. Overfitting was a problem with the current VGGNet-16 model due to its 16 layers. Consequently, the VGGNet-12 model was highly recommended by the researchers for

breast cancer detection [24]. The VGGNet-16 model was affected by the overfitting problem in the dataset used for breast cancer classification. After discovering that the original VGGNet-16 model was overfitting, the researchers decided to lower the model's total number of layers[25]. Following the development of more VGGNet models, including VGGNet-13 and VGGNet-19, the study published a revised version of the VGGNet model called VGGNet-12 as a recommendation to their research [26]. Using the breast cancer dataset, the researchers evaluated the model's effectiveness in comparison to CNN and LeNet. Compared to the model used in this investigation, the proposed VGGNet-12 model achieved a lower accuracy score of 0.85 according to the simulation data. In many breast cancer detection studies, the proposed VGGNet-12 model achieved better results than the state-of-the-art methods [27], [129].

Table 2.2 Summary of the Models' Limitations

Literature References	Techniques used	Remarks	Research Gaps
Panigrahi et al., 2020 [130]	Preprocessing, Feature Extraction, Decision Tree, K-Nearest Neighbor, Random Forest, and Support Vector Machine, Training and Testing, Performance Evaluation	The paper discusses the implementation and evaluation of various machine learning techniques suitable for plant leaf disease detection and classification. The effectiveness and accuracy of the crop leaf disease detection system are affected by the algorithms used, so their selection is critical.	Dataset availability, feature extraction and selection, ensembling, Image gradient, and principal component analysis.
Wang et al., 2021[131]	Traditional Machine Learning Algorithms such as Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), Random Forests, and Decision Trees. These algorithms are widely used and serve as benchmarks for comparison.	The paper focuses on evaluating the performance of image classification algorithms rather than proposing novel algorithms. The purpose of this research is to help researchers and practitioners choose the best algorithm for image classification tasks by comparing the performance of deep learning and traditional machine learning algorithms. The researchers used multiple image datasets for evaluation, ensuring a diverse range of	The study focuses on a limited set of traditional machine learning and deep learning algorithms. The study does not extensively discuss techniques like fine-tuning or transfer learning, which can enhance the performance of deep learning models. The computational requirements of the algorithms are not extensively discussed.

		images and addressing the problem of algorithm bias. The selection of appropriate datasets is crucial to ensure the generalizability of the findings. The authors use standard evaluation metrics, including accuracy, precision, recall, and F1-score, to measure the performance of the algorithms. These metrics allow for a comprehensive assessment of the classification results.	To provide a more thorough evaluation of the algorithms, future studies should investigate the trade-off between computational efficiency and performance.
Wang et al., 2020[132]	Transfer Learning i.e Inception-V3, Xception, VGG-16, and Resnet-50, Ensemble Learning	The goal of the study is to improve the classification models and make them more accurate by using pre-trained models and combining multiple classifiers. The pre-trained model can capture general features from a large dataset, which can help extract relevant features for cervical histopathology image classification.	The study could discuss the experimental setup and evaluation metrics used to assess the performance of the proposed approach. This includes the reporting of important evaluation metrics such as accuracy, precision, recall, and F1-score. The study could address the generalizability and scalability of the proposed approach. To ensure the model can handle

		However, by integrating the predictions of various classifiers into one, ensemble learning helps to overcome the shortcomings of individual classifiers and boost overall performance. The classification accuracy and reliability were both improved by the ensemble models.	large datasets or real-world deployment scenarios, it is crucial to test how well the models function on a variety of datasets.
Farmonov et al., 2023[133]	Computer Vision, Hyperspectral Imaging, Feature Extraction, and Machine Learning Techniques; Convolution Neural Networks, Random Forest, and Support Vector Machine.	The paper explores different feature extraction methods, such as PCA, MNF, and SAM. These methods are commonly used in hyperspectral data analysis, and their application in crop type classification helps in enhancing the classification accuracy by capturing relevant spectral information. The study compares the performance of different machine learning algorithms, including SVM, RF, and CNN. By evaluating and comparing the results obtained from each algorithm, the paper provides	The study does not explicitly mention the specifics of the dataset used for classification and the validation methodology employed. It would be helpful to have details on the size, diversity, and geographic distribution of the dataset, as well as information on how the classification accuracy was validated. The study focuses on crop type classification using DESIS hyperspectral imagery. It does not, however, address the question of whether or not the proposed method is applicable

		insights into the suitability and effectiveness of these algorithms for crop type classification using hyperspectral data.	to other hyperspectral datasets or other geographical regions. Further research could explore the transferability of the developed models to different datasets and regions. The study does not delve into the details of hyperparameter optimization for the machine learning algorithms used. The selection and tuning of these algorithms' hyperparameters may impact their performance. The effect of hyperparameter optimisation on the accuracy of classifications should be the subject of future studies.
Alem et al., 2022[134]	Preprocessing, training, evaluation, and deep learning models, which include Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs).	The paper provides a comprehensive evaluation of deep learning models for remote sensing image classification, which can help in selecting the appropriate model for specific applications. High accuracy in remote sensing image classification	The study does not explore the challenges and limitations of using deep learning models for remote sensing image classification. Understanding these limitations can guide future research in addressing these issues. The researchers don't provide any new methods or

		tasks can be achieved by using deep learning models, as shown by the researchers. The use of preprocessing techniques such as resizing, normalization, and data augmentation enhances the performance of the deep learning models. The comparative analysis provides insights into the strengths and weaknesses of different deep learning models, aiding researchers and practitioners in choosing the most suitable model for their specific requirements.	advancements to the area of remote sensing image classification; instead, they focus on assessing the effectiveness of current deep learning models. The paper does not discuss the interpretability of the deep learning models. For remote sensing applications to be transparent and to provide insights into decision-making, interpretable models are important.
Yin et al., 2020[135]	Extreme Learning Machine (ELM), Comprehensive Evaluation Model, Cumulative Variation Weights, Weighted Fusion Strategy	The technique for extracting features from hyperspectral images is not discussed in the article. It assumes that the hyperspectral data has been preprocessed and reduced to a feature representation suitable for classification.	The study does not explicitly mention the specific techniques used for feature extraction and preprocessing of hyperspectral images. The choice of feature extraction method can significantly impact the classification performance. Future research could focus on

		The proposed method is evaluated on three benchmark hyperspectral datasets: Indian Pines, Pavia University, and Salinas. The authors evaluate their method in comparison to other classification techniques, such as ELM-based approaches, Random Forest (RF), and Support Vector Machine (SVM). By outperforming the baseline approaches in terms of classification accuracy, the experimental results illustrate that the comprehensive evaluation model and weighted fusion procedure that was proposed are advantageous.	exploring and comparing different feature extraction techniques specifically tailored to hyperspectral data. A more comprehensive comparison with a wider variety of hyperspectral image classification algorithms will greatly enhance the study's value. Currently, it only compares the proposed approach with a few state-of-the-art methods. To fully grasp the merits and shortcomings of the proposed approach in contrast to other advanced techniques, more comparative evaluations are necessary.
Rui et al., 2020[136]	BBO-HS algorithm- The BBO-HS algorithm is a hybrid algorithm that combines the Biogeography-Based Optimization (BBO) algorithm with the Harmony Search (HS) algorithm	The proposed BBO-HS algorithm is used to optimize the SVM classifier, aiming to improve the classification performance for medical CT images. The paper claims that the BBO-HS algorithm enhances the SVM classifier's	The study focuses on the BBO-HS algorithm for optimizing the SVM classifier. It would be beneficial to compare the performance of the proposed method with other optimization techniques commonly used in image classification tasks, such

	Support Vector Machine (SVM)	capability to handle complex medical image data and improve the accuracy of classification. The paper evaluates the performance of the proposed model using two metrics: classification accuracy and mean square error (MSE). Classification accuracy signifies the proportion of CT images that were correctly classified, while MSE quantifies the disparity between the anticipated and real class classifications.	as genetic algorithms, particle swarm optimization, or ant colony optimization. The study does not provide detailed information about the dataset used for the experiments. To evaluate the proposed model's generalisability, it would be helpful to provide the dataset's properties, such as the number of CT scans, classifications, and any preprocessing processes. While accuracy is an important metric, evaluating the robustness and interpretability of the proposed model is also crucial in medical imaging applications. Research areas that could provide useful results include assessing the model's performance on multiple dataset subsets, evaluating its sensitivity to input alterations, and providing interpretability of the learnt features.

Deepa et al., 2022[137]	The Maximum A Posteriori (MAP) Firefly Algorithm, Pattern Descriptors Orientation. The MAP Firefly Algorithm and the pattern descriptors orientation approach are used to form a hybrid machine learning algorithm.	This research presents a new method for classifying brain pathologies using the MAP Firefly Algorithm in combination with pattern descriptors orientation.This combination of techniques has the potential to enhance the performance of traditional machine learning algorithms. The proposed hybridized machine learning algorithm offer improved accuracy in brain pathology classification, which can have significant implications for medical diagnosis and treatment planning. The paper provides sufficient details about the techniques used, allowing researchers to reproduce the experiments and verify the results.	Accuracy, sensitivity, specificity, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC) are some of the measurements that need more clarification. Details such as the size of the dataset, acquisition parameters, and ground truth labeling process would be helpful for assessing the generalizability of the proposed algorithm It would be valuable to investigate how the proposed approach contributes to the interpretability of brain pathology classification, as interpretability is crucial in medical decision-making. It would be beneficial to conduct validation studies using real-world clinical data to assess the algorithm's performance in a practical setting.

Priyadharshini et al., 2021[138]	Optimisation, Data Augmentation, Convolutional Neural Networks (CNNs), Transfer Learning	The study needs to delve into the measures that were used to measure how well the deep learning model that was proposed performed. Area under the receiver operating characteristic curve (AUC-ROC), recall, accuracy, precision, and F1 score are common metrics for image classification. The problem and dataset at hand should be considered while selecting evaluation metrics.	The study may identify potential research gaps related to the exploration of new deep learning architectures or modifications to existing architectures to improve image classification performance. These gaps could be related to the network design, feature extraction, or model optimization techniques.
Wan et al., 2021[139]	Dimensionality Reduction, Feature Extraction, Machine Learning classification; Support Vector Machine (SVM), Random Forest (RF), and K-Nearest Neighbours (KNN)	The primary goal of this research is to provide a reliable and effective classification system for detecting underground metal targets. The proposed methodology aims to accurately classify metal targets while minimizing computational complexity. The authors compare the effects of three dimensionality reduction methods on	It would be beneficial to evaluate the proposed methodology on a wider range of datasets, including different geographical locations, soil conditions, and metal target characteristics. This would help assess the robustness and generalizability of the approach. The study focuses on extracting statistical features after

		classification performance: Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Independent Component Analysis (ICA). This study is useful for learning about the advantages and disadvantages of each method.	dimensionality reduction. Exploring additional feature selection techniques or investigating domain-specific features that can capture more discriminative information about the metal targets could enhance the classification performance.

2.6 Model Implementation Approaches

When it comes to applications ranging from computer vision to natural language processing, the subfield of machine learning known as deep learning has been a huge success. The multi-stage process of implementing a deep learning model starts with data preparation and continues with training and evaluation of the model. In this research, we explore the finer points of putting into practice a deep learning model, touching on crucial steps and factors [20], [140].

i. Data Preparation

Data preparation is a crucial initial step in deploying a deep learning model. To successfully train accurate models, high-quality, carefully managed datasets are required. Data cleansing, normalization, and enhancement are all part of this process. By eliminating inconsistencies and fixing mistakes, data cleaning creates trustworthy input for the model. Normalization standardizes data, bringing it to a constant size, which is critical for the convergence of deep neural networks. Data augmentation techniques are used to provide new training samples that enhance the model's ability to generalize [21], [141].

ii. Model Architecture Design

The selected architecture has a significant impact on the effectiveness of a deep learning model. Researchers and practitioners alike need to pay great attention to selecting or creating structures that are optimal for the given purpose. The processing of sequence data is best handled by Recurrent Neural Networks (RNNs), while image-related tasks are more often handled by Convolutional Neural Networks (CNNs). Layers, activation functions, and neuronal connections are all part of the architecture's design process. TensorFlow and PyTorch are two examples of frameworks that make it easy to build and modify such models [24], [142].

iii. Hyperparameter Tuning

Hyperparameters are part of deep learning models that have a major effect on the model's performance but are not learned during training. Regularization parameters, batch sizes, and learning rates all fall under this category. In order to get the best results out of the model, hyperparameter tuning is an essential process. Finding the best

hyperparameters is a tedious process, but methods like grid search and random search may help. Improving model performance requires careful hyperparameter tuning [26], [143].

iv. Model Training

The trained model is fed the cleaned data, and its parameters are fine-tuned with the help of this data. Stochastic gradient descent and other optimization techniques are often used for this task. To train the model to make predictions, it is necessary to tune its internal weights. Training is repeated until no more improvements in performance can be achieved by the model. To ensure the model is learning well at this stage, it is essential to monitor parameters like loss functions and validation accuracy [28], [144].

v. Evaluation and Validation

After the model has been trained, its ability to generalize must be tested on new data. Evaluation metrics vary from the mean squared error for regression tasks to accuracy, precision, and recall for classification tasks. When working with validation datasets, it is essential to fine-tune the model to prevent overfitting, a situation in which the model exhibits good performance on the training data but degrades when subjected to new, unknown data [30], [145].

vi. Deployment and Inference

The deep learning model is deployable after it has been trained and validated. The model must be deployed so that it may be used to analyze new data in a practical setting. Inference, or making predictions on new data using the trained model, is an important component of deployment. An important part of deployment is inference, or the act of utilizing the trained model to generate predictions on new data. Optimizations such as model quantization and compression are typically necessary for fast deployment so that the model can operate in real time on a variety of devices [32], [146].

vii. Considerations and Challenges

When putting into action a deep learning model, there are a number of factors to think about and obstacles to overcome. Training huge models on massive datasets is one problem that often requires a lot of processing power. The difficulty of understanding

the inner workings of deep neural networks raises further concerns about the interpretability of models. To guarantee the ethical deployment of deep learning models, it is essential to address challenges like data bias and ethical concerns [35], [147].

It is clear that there are a lot of steps to implementing a deep learning model. It is impossible to create efficient and trustworthy deep learning models without paying close attention to detail at every stage. Deep learning has the potential to solve complex real-world issues, but researchers and practitioners need to stay updated on the newest advancements and best practices for the technology [148], [149].

Table 2.3 Deep Learning Model Implementation Approaches

S/N	Approach	Description
1	Framework Selection	Choose suitable frameworks (e.g., TensorFlow, PyTorch, Keras) based on task complexity and support.
2	Model Architecture	Select appropriate layers, connections, and activation functions for the problem type.
3	Data Preprocessing	Prepare data through cleaning, normalisation, augmentation, and handling missing values.
4	Data Augmentation	Increase dataset diversity using transformations to improve generalisation and reduce overfitting.
5	Transfer Learning	Use pre-trained models and fine-tune them when labelled data is limited.
6	Hyperparameter Tuning	Optimise parameters such as learning rate and batch size for better performance.
7	Regularization	Apply methods like dropout and L1/L2 regularisation to prevent overfitting.
8	Optimization Algorithms	Use suitable optimisers (e.g., Adam, SGD) for stable and efficient training.
9	Loss Function Selection	Choose appropriate loss functions based on task requirements (e.g., classification or regression).
10	Model Training	Train models using backpropagation while monitoring validation performance.
11	Evaluation Metrics	Assess performance using accuracy, precision, recall, and other relevant metrics.
12	Deployment Considerations	Plan deployment based on hardware, speed, and system integration requirements.
13	Explainability	Apply methods such as SHAP and LIME to improve model interpretability.
14	Continuous Monitoring	Regularly monitor and update models as data and performance change.
15	Ethical Considerations	Ensure transparency, fairness, and responsible Artificial Intelligence implementation.

2.7 Model Validation Approaches

As a check on the model's generalization abilities, validation is an essential part of deep learning model development. To ensure that models are accurate and work as expected in the real world, there are a number of approaches available for validation. Various deep learning model validation approaches are analysed in this research, along with their applications and complexities [150], [151].

i. Train-Test Split

A common technique for validating results is the train-test split, which involves dividing the dataset into two parts: one for training and one for testing. Although this method is straightforward, it has limitations, especially when dealing with small datasets, since the model's performance could be highly reliant on the specific random split. It establishes a standard against which models may be evaluated, although it may not be representative of the data as a whole [32], [152].

ii. K-Fold Cross-Validation

In K-Fold Cross-Validation, the dataset is divided into k folds, the model is trained k times, and it is evaluated on different folds each time. This approach improves the evaluation of the model's performance by using the majority of the data for both testing and training. The model has to be trained k times, which increases the computing cost, particularly when dealing with large datasets [36], [153].

iii. Stratified Cross-Validation

Stratified Cross-Validation ensures that the class distribution remains consistent across all folds, which helps address issues of class imbalance. This is especially critical for classification tasks because some classes may contain less data. This method improves model assessment by giving a more representative picture of the model's performance across multiple classes by preserving class balance in both the training and test sets [39], [154].

iv. Leave-One-Out Cross-Validation (LOOCV)

Large-Order Cross-Validation (LOOCV) is the cross-validation method used when the dataset size is equal to the number of folds (k). The model is trained on some of

the data, and then tested on some other data, in each cycle. Although LOOCV is computationally costly, particularly when dealing with large datasets, it gives an impartial measure of model performance. It is often used when there are few observations available and everyone counts [40], [155].

v. Time Series Cross-Validation

If your data is ordered chronologically, Time Series Cross-Validation is the method for you. Data is sequentially divided into training and testing sets to simulate real-world scenarios where predictions are formed based on upcoming observations. Applications where the sequence of data is important, such as financial forecasting or climate prediction, may benefit greatly from this method [41], [156].

vi. Nested Cross-Validation

Model assessment is handled by an outer loop of K-Fold Cross-Validation, while hyperparameter tuning is handled by a second, inner loop, in Nested Cross-Validation. This method is useful for improving model hyperparameters because it prevents hyperparameters from being overfit to a single dataset. An improved prediction of the model's behavior on unknown data is achieved by repeatedly picking new training and validation sets [42], [157].

It is crucial to carefully examine the validation strategy to use while creating deep learning models. There are benefits and drawbacks to each method, and choosing one relies on a number of criteria, including the size of the dataset, the distribution of classes, and the time dimension of the data. To ensure the stability and generalizability of deep learning models in a wide range of practical contexts, it is critical to use and comprehend these validation procedures. If researchers and practitioners are serious about advancing deep learning and getting reliable insights into their models' performance, they should closely consider these techniques [158], [159].

Table 2.4 Model Validation Approaches

S/N	Approach	Description	Advantages	Considerations
1	Train-Test Split	Divides data into training and testing sets for model development and evaluation.	Simple, fast, and suitable for large datasets.	Results may vary depending on the random split.
2	K-Fold Cross-Validation	Splits data into k folds, training on k-1 folds and validating on the remaining fold repeatedly.	Provides reliable performance estimates and reduces variability.	Computationally expensive for large datasets.
3	Stratified Cross-Validation	Maintains class distribution across different validation folds.	Effective for imbalanced datasets and ensures class representation.	Requires more computation than standard k-fold.
4	Leave-One-Out Cross-Validation	Uses one sample as validation while training on all remaining samples.	Provides unbiased evaluation and works well for small datasets.	Very computationally expensive for large datasets.
5	Time Series Split	Splits data chronologically for time-dependent prediction tasks.	Mimics real-world future prediction scenarios.	Sensitive to temporal patterns and data changes.
6	Holdout Validation Set	Separates validation data for hyperparameter tuning alongside training and testing sets.	Prevents test-set contamination and improves tuning.	Requires sufficient data volume.
7	Bootstrapping	Creates multiple samples with replacement to estimate model performance.	Handles performance variability and improves robustness.	Computationally intensive and not suitable for all datasets.

2.8 Research Gaps

Despite significant advances in deep learning research in recent years, some key gaps remain that prevent the technology from reaching its full potential. One area where deep learning models fall short is in their interpretability and explainability. Although these models function well, they are typically black boxes, making it difficult to interpret how they arrive at their conclusions. This lack of openness is cause for alarm, particularly in settings where trust and responsibility are crucial, like the medical or financial sectors. To close this gap and encourage wider use, researchers need to figure out how deep neural networks make their decisions [160], [161].

Additionally, efficiency and scalability remain constant obstacles in deep learning studies. Training more complex and larger models consumes a lot of time, effort, and energy as their size and complexity increase. This makes it more difficult to deploy them, especially in places with limited resources. Research efforts need to concentrate on creating more efficient training algorithms, model structures, and hardware solutions to make deep learning more scalable and economically sustainable[162], [163].

The extent to which deep learning models are able to generalise is another ongoing concern. These models often do well on training data, but they may not perform so well when subjected to real data. To bridge this knowledge gap, researchers need to learn more about model generalization and create techniques to improve model resilience across different types of data [164], [165].

Interpretability, adversarial robustness, scalability, efficiency, and generalizability are all areas where further study is needed in deep learning. Tackling these issues will not only propel the discipline forward but will also pave the way for the ethical and efficient rollout of deep learning technologies in a wide range of settings[166], [167].

2.9 Summary

The literature review sheds light on the considerable advances achieved in the use of deep learning models for decision support in the field of crop disease and pest management. Reviewing this amount of literature highlights the possibility that these models will revolutionize stakeholder decision-making in agriculture. There is hope that early detection, diagnosis, and mitigation measures for crop diseases and pests can be improved with the incorporation of deep learning techniques into existing crop management systems.

Images, sensor data, and historical records are just a few examples of the many datasets that have been analysed in the literature by using contemporary technologies like Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). To help stakeholders make informed decisions, these models can identify complex patterns and provide valuable information. Collectively, the literature review adds to the expanding body of knowledge demonstrating the usefulness of deep learning in handling the complex issues provided by crop diseases and pests.

Although the study demonstrates potential for using deep learning models to make decisions about crop diseases and pests, it also brings attention to ongoing challenges. Stakeholders frequently need a clear grasp of the decision-making process to trust and accept such systems, hence the interpretability and explainability of these models arise as significant considerations. This gap must be closed if deep learning technologies are to be successfully adopted by the agriculture industry.

Furthermore, the research stresses the need for contextually adjusted and environment-robust models. Deep learning models have many potential applications in agriculture, but researchers are still trying to figure out how to make them more robust. Furthermore, it is necessary for the broad use of these decision support systems to address concerns of data availability and quality in particular areas.

The literature study found that stakeholders in crop disease and pest management will benefit greatly from decision support systems that use deep learning models. The study presents some encouraging results, but it also brings to light several issues that require further attention; it also emphasizes the need for multidisciplinary efforts to improve

the robustness, interpretability, and contextual adaptation of these models in practical agricultural contexts.

CHAPTER THREE
METHODOLOGY

3.1 Introduction

The methods that were used to achieve the research objectives are described in this chapter. This research used an experimental research design to develop and test a deep learning model for decision support of stakeholders in crop diseases and pest management. The researcher used the Kaggle website to access the secondary datasets containing images of crop diseases and pests. To improve the model's flexibility and applicability, the images were first resized to 224×224 pixels, then their pixel values were normalized. To further improve the model, data augmentation methods such as random rotations, flipping, and zooming were used. The proposed model, which combined Convolutional Neural Network (CNN) for automatic feature extraction and Support Vector Machine (SVM) for the final classification, was developed in Python using the PyTorch deep learning framework. Models such as DenseNet, AlexNet, EfficientNet, VGG, and a baseline CNN were used for comparison purposes. To assess the model's accuracy, precision, recall, F1-score, and confusion matrix, the researcher looked at its reliability and its ability to generalize across different types of crop diseases and pests. Adherence to ethical issues was ensured by conducting research using only open-access datasets, properly attributing sources, being transparent in reporting, and keeping integrity at all times.

3.2 Research Philosophy

The main set of principles that directs the conduct of research is known as a research paradigm or research philosophy. The four most prominent schools of thought in the field of research are pragmatism, interpretivism, positivism, and realism. In positivism, the study of phenomena is characterized by an emphasis on measurement and the provision of evidence. Experiments provide fundamental conclusions based on data obtained, test hypotheses, and demonstrate the link between dependent and independent variables. They also give formal statements. Thus, the goals of this study were accomplished through the use of a pragmatic research philosophy[168], [169].

The investigation of a deep learning model for use in decision support for crop disease and pest management was well-suited to a pragmatism research philosophy.

Pragmatism is an approach to problem-solving that prioritizes action above theory. To find a workable answer to this research question, a realistic strategy that takes both theoretical and practical factors into account in equal measure was used. Researchers adopting a pragmatic attitude will likely emphasize the creation of a deep learning model that not only corresponds with current theoretical underpinnings in machine learning but also solves the unique demands and constraints encountered by stakeholders in agriculture[170], [171].

Farmers and agronomists all had different needs, which were taken into account by the pragmatism research philosophy. The researcher adhered to this philosophy in order to conduct a comprehensive analysis of the deep learning model's factors, like the model's ease of use, interpretability, and scalability. This method conformed to the pragmatic idea of what works by ensuring that the developed model was successful in both theoretical and practical settings[172], [173].

A pragmatic research approach made it easier to modify methods and procedures in light of empirical data and stakeholder input. By iterating and adapting, the researcher made sure that the deep learning model kept up with the demands of the agricultural sector and ultimately developed a better decision-support model. The pragmatic research philosophy laid the groundwork for the development of a deep learning model that was both theoretically sound and practically useful in aiding stakeholders in the control of crop diseases and pests[174], [175].

3.3 Research Design

The research process relied on a conceptual framework for collecting and analysing data. The investigation of a deep learning model for use in decision support for crop disease and pest management benefited greatly from a mixed-methods research design. This methodology combined both quantitative and qualitative research approaches, allowing for a thorough and holistic inquiry. The quantitative component comprised the development and assessment of the deep learning model, leveraging performance measures like accuracy, precision, recall, and F1-score. This allowed for a quantitative evaluation of the model's utility as a tool for making decisions. To systematically examine the effect of the deep learning model on decision-making outcomes, an experimental research design, a subset of quantitative design, was

studied. Finding a cross-section of relevant parties, such as farmers and agronomists, to take part in the research was the first step. To rule out confounding variables, it was essential to randomly assign participants to the control group[176], [177].

The control group continued with their current decision-making processes, while the experimental group was trained on how to utilize the deep learning model for decision support in crop disease and pest management. After that, both sets of people were shown hypothetical examples involving crop diseases and pests, and their responses were recorded and compared. The efficacy of the deep learning model was statistically evaluated based on performance criteria including accuracy, timeliness, and efficiency of decisions[178], [179].

Concurrently, interviews, questionnaires, and focus groups with stakeholders, including farmers and agronomists, made up the qualitative component. Implementing the deep learning model in real-world agricultural contexts has a number of contextual complexities, usability problems, and practical implications that were addressed by the qualitative data. [24], [180].

3.4 Research Process

This research process sets out the steps used to accomplish each of the research objectives. In order to accomplish objectives one to four, the process included four primary steps. The first objective was achieved by collecting data from both the literature review and experiments to determine the deep learning model factors that influence decision support in crop diseases and pest management. The second objective was addressed in the second step by conducting experiments to develop an enhanced deep neural network model for accurate image classification. The third objective was addressed by creating a user-friendly tool for end-users to interact with the model. The last step of the research process was to validate the developed model to affirm that it provides accurate decisions [181], [182]. Figure 3.1 below illustrates the steps of the research process.

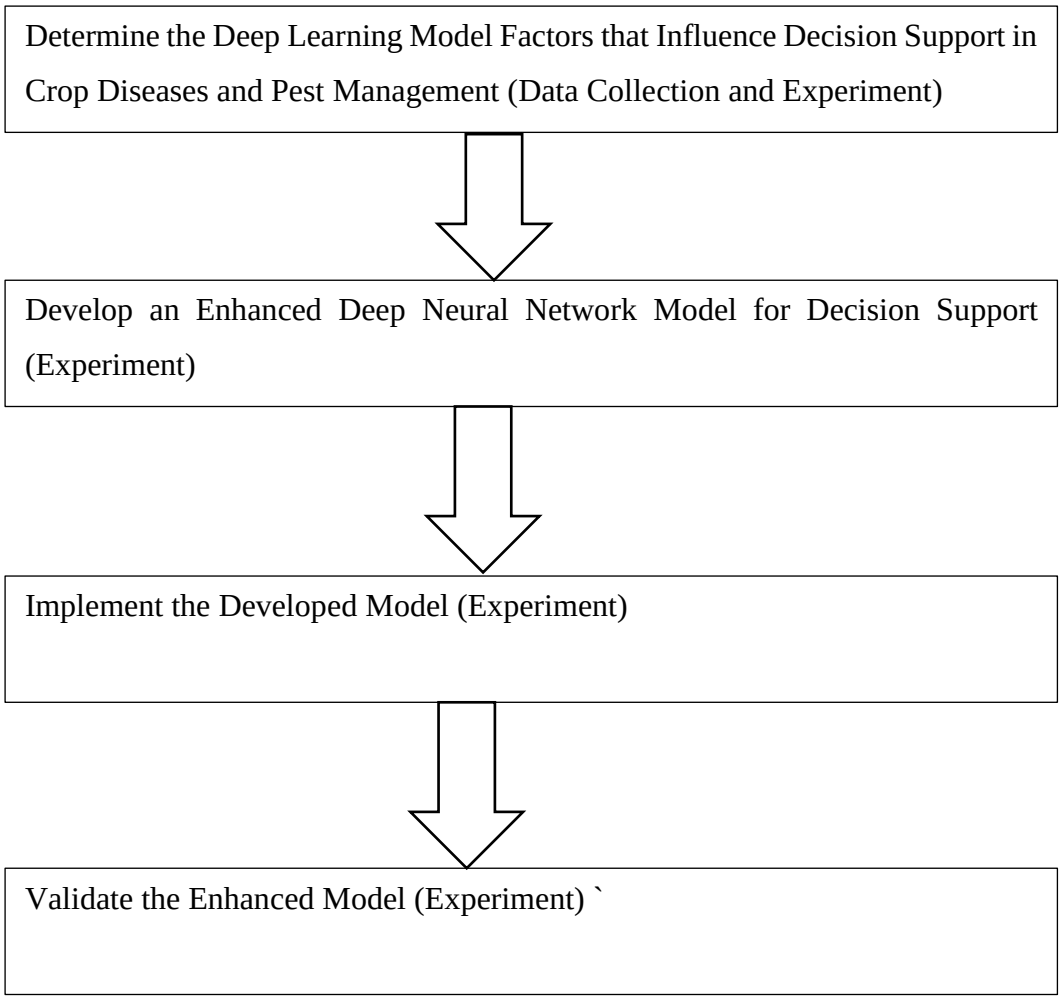


Figure 3.1 Research Process

Using a variety of approaches, the study set out to accomplish four main goals. The first step was to do a systematic literature review and experiments to identify and analyze deep learning model factors influencing decision support in crop diseases and pest management. As part of this process, the researcher had to search through scholarly journals and research to find the most pertinent information. Second, to increase accuracy in classifying crop diseases and pest infestations, the researcher used a design science method to develop an enhanced deep neural network model[183], [184]. This model used a Convolution Neural Network for feature extraction and a Support Vector Machine for classification. Third, a user-centered design method was used to integrate the developed model into an intuitive software platform. Stakeholders were given access to a software platform, including image uploading, classification results, pesticide recommendations, and agrovet locations through the

software platform. The model was tested and compared to other models using real-world datasets to ensure it was robust and generalizable. These benchmark models included DenseNet, AlexNet, EfficientNet, VGG, and CNN. The model's adaptability to a wide range of crop diseases and pest varieties was further tested by performance metrics such as accuracy, precision, recall, F1 score, training, and validation loss [185], [186].

Table 3.1 Strategies to Achieve Specific Objectives

S/No.	Objective	Method/Strategy	Description
1	Identify factors influencing deep learning-based crop disease and pest decision support.	Systematic Literature Review and Experiment	Reviewed studies to identify key factors and validated their impact through deep learning model experiments.
2	Develop an enhanced deep learning model for crop disease and pest classification.	Design Science	Designed and improved a neural network model using techniques such as CNN and SVM for accurate classification.
3	Implement the model into a user-friendly decision-support platform.	Software Development	Integrated the model into a platform supporting image upload, disease classification, recommendations, and agrovet services.
4	Validate model robustness and generalisation in real-world agricultural settings.	Experiments and Comparative Analysis	Compared the model with benchmark approaches (DenseNet, AlexNet, EfficientNet, VGG, CNN) using accuracy, precision, recall, F1-score, and loss metrics.

3.5 Description of the Training and Test Data Set

The development of a deep learning model to assist with crop disease and pest management decision-making relied heavily on the careful selection of training and test datasets. The training dataset was made up of a wide variety of labeled images showing both diseased and healthy crops, as well as pests. The secondary dataset was obtained from Kaggle with labeled images consisting of a variety of classes, and it was used for both training and testing purposes. The deep learning model used these images as training data to identify and generalize the features and traits associated with various crop diseases and pests. The training dataset's diversity was essential for the model to

get a thorough comprehension of the wide variety of conditions it faced in practical agricultural contexts [187], [188].

The test dataset, on the other hand, was an entirely new group of images that the deep learning model never saw while it was being trained. This data was used to evaluate how well the model could generalize to new conditions and provide accurate predictions. It was important to incorporate a diverse selection of images that accurately depict the complexity and variance of real-world agricultural fields in the test dataset. The researcher objectively evaluated the model's efficacy in recognizing and classifying crop diseases and pests under unforeseen circumstances by utilizing a separate test dataset [189], [190].

The success and reliability of the deep learning model depended on both the training and test datasets. In order to make the model more flexible, it was necessary to update and expand these datasets on a regular basis to account for new information, new crop health issues, and new agricultural practices. For these datasets to be transparent and reproducible, the images were well documented and annotated so that other researchers may check the results and build upon the work. A deep learning model that helps stakeholders make informed decisions about crop diseases and pest control relies heavily on the careful creation and administration of training and test datasets [191], [192].

In order to classify crop diseases and pests, a hybrid deep learning model was first trained using collected data. The model's performance was substantially impacted by the dataset's variety and quality. Publicly accessible agricultural databases and agricultural extension organizations provided the data used in this study. Both PlantVillage and Kaggle contributed to the solid foundation of the dataset by providing extensively labeled images of various crop diseases and pests. The Kaggle repository provided the dataset that was used for this study. Table 3.2 shows the distribution of images for maize and tomato crops under different health conditions and disease states. There were a total of 7,316 images of maize leaf diseases used. Among them, 1,908 showed healthy maize, 1,907 common rust, 1,859 leaf spots, and 1,642 northern leaf blight. The tomato disease dataset included bacterial spots, early blight, mold,

spider mites, target spot, mosaic virus, and yellow leaf curl virus, and each category contained 1000 images, resulting in a total of 10,000 images for tomatoes [193], [194].

Table 3.2 Crop Disease Image Distribution

Crop	Disease	Number of Images
Maize	Northern Leaf Blight	1908
	Common Rust	1907
	Healthy	1859
	Gray Leaf Spot	1642
Total Images for Maize		7316
Tomato	Bacterial Spot	1000
	Early Blight	1000
	Healthy	1000
	Late Blight	1000
	Leaf Mold	1000
	Septoria Leaf Spot	1000
	Spider Mites	1000
	Target Spot	1000
	Mosaic Virus	1000
	Yellow Leaf Curl Virus	1000
Total Images for Tomato		10000

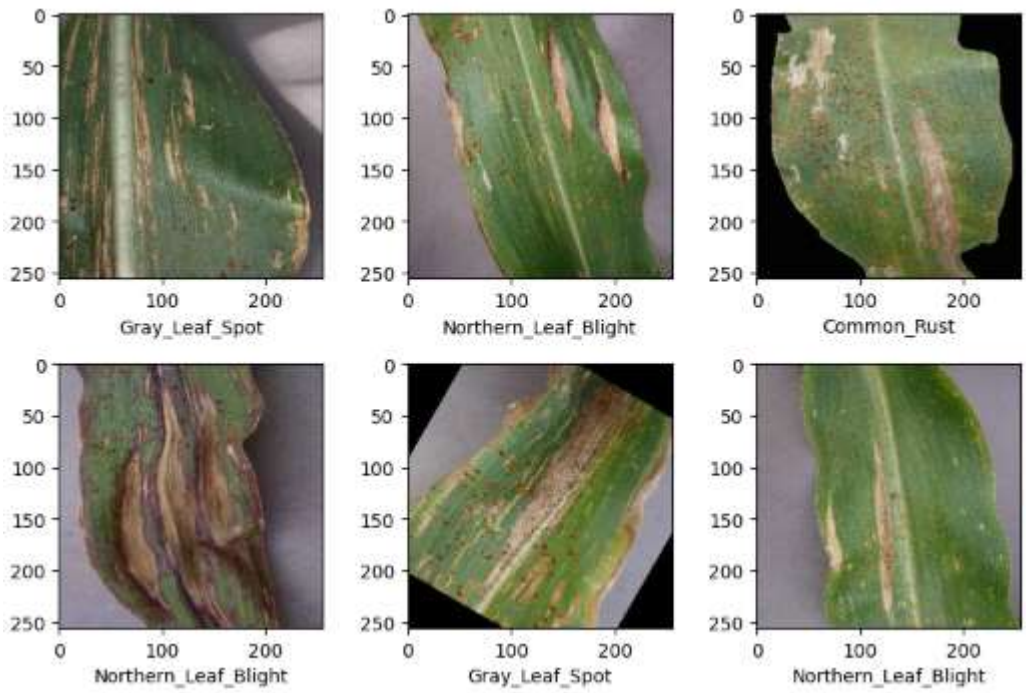


Figure 3.2 Sample of Maize Leaf Disease Images

A visual representation of six images of maize leaves, identified according to the disease affecting each, is shown in Figure 3.2. Gray Leaf Spot, Northern Leaf Blight, and Common Rust are the three primary types of maize leaf diseases shown in the illustration. The top left and center bottom images show gray leaf spots, which are elongated, thin, tan lesions on the leaves. The boundaries of these spots are well-defined and run along the veins. The three images of Northern Leaf Blight show big, circular lesions that range in color from gray to brown and are arranged lengthwise on the leaves. The tiny, reddish-brown pustules that cover the leaf surface are an indicator of Common Rust, as seen in the top right image. The images cover a range of orientations, lighting, and backgrounds, indicating an attempt to include a varied set of conditions for deep learning model training and validation. The pixel dimensions (0–250) are shown on the axis of each image in Figure 3.2, and the variety of images captured in the maize leaf disease dataset helped to build a more robust and generalizable classification model [44], [195],[196].

In Table 3.3, you can see all the information on the crop pests, including how many images of each type of pest. More specifically, there were 390 earwigs, 405 snails, and 400 ants in the images. Weevils were 394, slugs 316, beetles 331, whereas wasps were 392. In contrast to the 246 earthworm images, there were 397 moth images. In addition, there were 329 images of caterpillars, 405 of bees, and 390 of grasshoppers. In Table 3.3, you can see a detailed visual depiction of these typical pests, with 4,395 images across all categories.

Table 3.3 Image Distribution for Pests

Pests	**Number of Images**
Ants	400
Snail	405
Earwig	390
Slug	316
Weevil	394
Wasp	392
Beetle	331
Earthworms	246
Moth	397
Bees	405
Caterpillar	329
Grasshopper	390
Total Images for Pests	**4395**

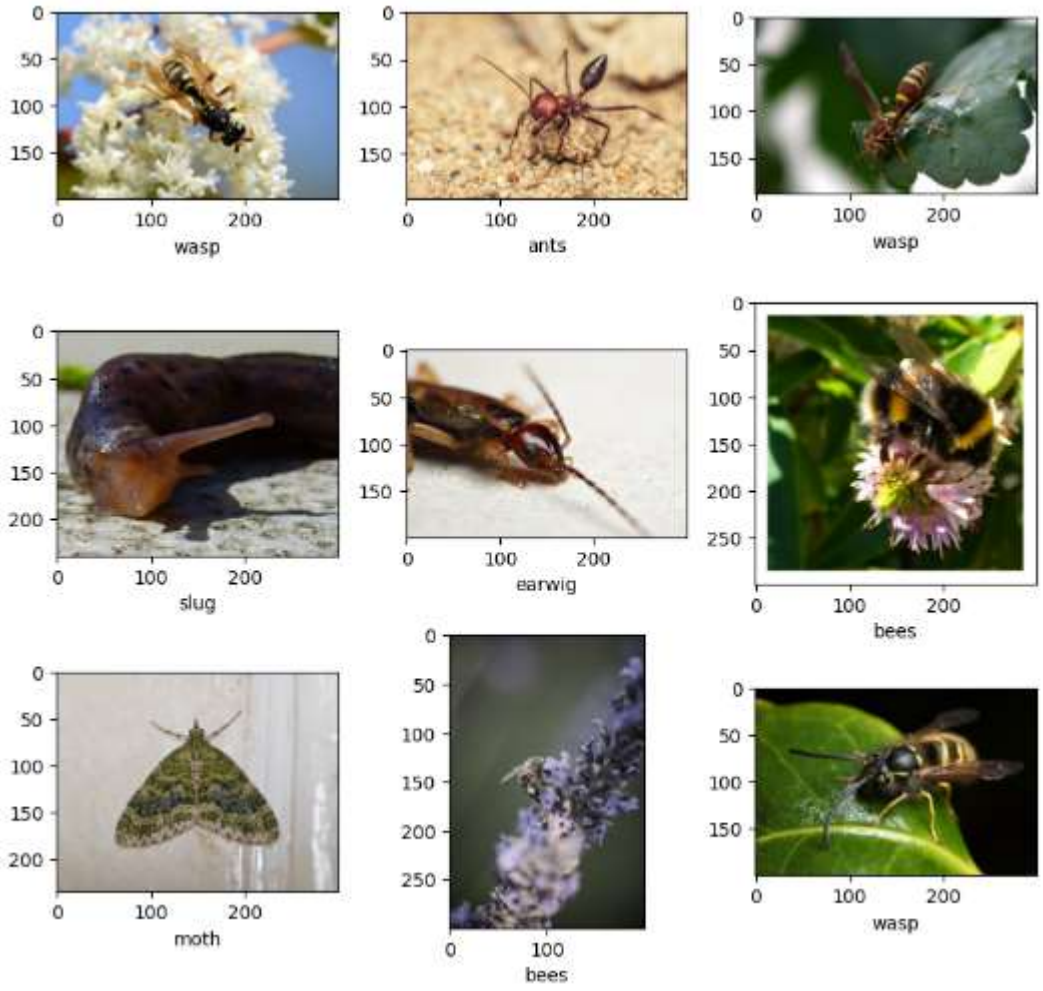


Figure 3.3 Sample of Pests Images

Nine images organized in a 3x3 grid provide an example of a pest image dataset, as shown in Figure 3.3. The names of the pests seen in each image are provided at the bottom. A wasp, some ants, and yet another wasp make up the top row, illustrating diversity within the category of pests. A beehive, an earwig, and a slug occupy the center row. The earwig has a slim shape and visible pincers, in contrast to the slug's soft, elongated body. The bee captured on a flower reveals characteristics that help identify its type of pollination. A moth, a bee, and a wasp make up the bottom row, highlighting the variation in lighting, background, and posture. The dataset included both harmful and useful pests, giving a wide array of visual cues that were used to train the deep learning model to classify pests. The dataset was well-suited for building a robust and accurate deep learning model for pest identification due to the images' varied angles, environment, and levels of focus [197], [198].

To guarantee that the collection was diverse, the dataset included images of a wide range of crop diseases and pest species. When training the model to generalize, it was essential to include a diverse set of images. With data annotation, the researcher trained the model on the correlations between visual features and their labels, ensuring that every image was correctly labeled. The images were enhanced using a variety of data augmentation methods. Rotating, inverting, cropping, and scaling are some of these methods. The model was reinforced, and the likelihood of overfitting was reduced using these methods by expanding and diversifying the dataset. By using a broad and high-quality dataset, this study provided the framework for creating an efficient deep-learning model for crop diseases and pest classification [199], [200].

3.6 Experimental Setup

The environment used to create the model was set up by creating an account with Google Colab, which is a platform-as-a-service integrated environment for running Python programs, and this enabled the researcher to buy a Tensor Processing Unit that was used to run the experiment. The collaborative and cloud-based features of Google Colab and Tensor Processing Unit, which provided access to powerful Graphics Processing Unit and Tensor Processing Unit resources, were major factors in this selection. With the platform's connection with Jupyter Notebooks, collaborative and interactive coding was made easier, and the setup process was simplified with pre-installed libraries. Tensor Processing Units improved efficiency by offering quicker model training times, making them ideal for deep learning tasks. Under Google Colab, a notebook application was created that allowed the researcher to type Python code as input and display the results of the executed code as output in a well-presentable environment. The following libraries were imported: Python Imaging Library for image processing, matplotlib and seaborn for data visualization, NumPy for data preprocessing, PyTorch, a deep learning library for computer vision, JSON to handle JSON files, os for operating system commands, random for generating random numbers, tqdm for status bar, and copy for copy functionality. To check if the Graphical Processing Unit for processing images was available, the Compute Unified Device Architecture PyTorch function was written to check the availability of the Graphics Processing Unit. The reproducibility of the code was also considered by setting the random seed to 3. The images to be used were loaded up using the helper

function and this was done first by creating an unnested list of file names, labels for each image in the list were determined, the count of the images in each crop disease and pest category were done, and the total number of train and test images for each crop disease and pest category were loaded and displayed using the helper function [201], [202].

3.7 Data Preprocessing

A robust deep learning model for crop disease and pest classification relied heavily on well preprocessed data for images of crop diseases and pests. The first step was to amass a large and varied dataset of images, with an even distribution of disease and pest types. After that, the evaluation of the model's efficacy was done by separating the data into training, validation, and testing sets.

To guarantee consistency across the dataset, it was crucial to resize all images to the same dimensions. Depending on the model architecture, the most common options for pixel size were 224x224 and 256x256. To improve the model's generalizability, data augmentation methods were used to enhance the variety of the training data by employing techniques like rotation, flipping, cropping, and adjustments to brightness or contrast.

Another crucial stage, normalization, accelerated model convergence during training. Particularly for RGB images, normalizing pixel values by making sure each channel has a mean of 0 and a standard deviation of 1 was crucial. One-hot encoding or label encoding was used to encode the class labels, which were essential for the model to comprehend and classify the various crop leaf diseases and pests.

Memory concerns were avoided, and enormous datasets were managed with the use of efficient data loading technology known as a data generator or pipeline. Oversampling, undersampling, or class weights were used to deal with class imbalances in the dataset, which ensured that the model did not unfairly benefit from the class with more images.

During data preparation, quality control was essential for finding and discarding outliers, mislabeled images, and damaged images. In addition, the researcher saved time and effort by making use of pre-trained deep learning models and then fine-tuning them for the specific task. In particular, for deep learning models, effective data

handling depended on having a well-structured data input pipeline. Finally, while evaluating the model, the researcher applied data augmentation to both the validation and testing sets [203], [204].

3.8 Loading the Dataset

The secondary dataset was obtained from Kaggle with labeled images consisting of a variety of classes. The images to be used were loaded up using the helper function and this was done first by creating an unnested list of file names, and labels for each image in the list which were determined, the count of the images in each crop disease and pest category was done, and the total number of train and test images for each crop disease and pest category were loaded and displayed using the helper function. The different categories of crop leaf diseases and pests were stored in two datasets, one for training and the other for validation.

The input and output values were initialized for images and labels, respectively, since the problem to be solved was a classification problem. The list of images was defined by loading the path and defining the working directory. The bias in the sequence of data was avoided by ensuring the input images were reshuffled before loading. The images were resized to a 224 x 224 size and extracted from their respective paths by looping over all the images. The labels and data lists were used to store the labels of the images, and the images acted as the input to the model. Convergence was achieved faster by normalizing the data after converting the lists of labels and data into NumPy arrays. The pickle file was used to store encoded class names, which were encoded by the use of labelbinarizer, and for class names to be used later during prediction, they were stored in a serialized manner. Finally, when the data was in the proper format, the train_test_split function was used to split the data into train and test sets. The function shuffled the data using the random_state parameter, the ratio of the split was determined using the test_size parameter, and also took the images with their labels as the parameter [205], [206].

3.9 Identification and Analysis of Deep Learning Model Factors Influencing Decision Support in Crop Diseases and Pest Management

The goal of this study was to use deep learning models to improve decision support for crop disease and pest management. To accomplish this, a two-pronged approach was used, including a systematic literature review and practical experiments. The Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) methodology was followed in the systematic literature review to ensure a clear and organized selection of academic publications from reputable databases such as Scopus, IEEE Xplore, ScienceDirect, and SpringerLink. This study looked at research that employed deep learning models for crop diseases and pest management, namely CNN, EfficientNet, DenseNet, ResNet, and VGG, etc. published between 2018 and 2024. Studies that enlightened on model performance, dataset features, evaluation criteria, and real-world applications were accessed using predetermined inclusion and exclusion criteria. To identify the factors that affected decision support systems' efficacy, important themes and factors were compiled, including accurate annotations, model depth, computational efficiency, adaptation to varied crop circumstances, and data quality [207], [208].

Algorithm: Systematic Literature Review on Deep Learning-Based Decision Support in Crop Diseases and Pest Management

Input: Published peer-reviewed articles on deep learning applications in crop disease and pest management

Output: Synthesized findings identifying and analyzing key influencing factors

Procedure:

1. **Review Planning**

   (a) The review objective was defined as identifying and analyzing key factors influencing decision support in crop diseases and pest management using deep learning models.

   (b) Research questions were formulated based on the PICOC framework (Population, Intervention, Comparison, Outcome, Context).

   (c) A review protocol was developed, outlining the inclusion/exclusion criteria, search strategies, and data extraction methods.

   (d) Relevant databases were identified, including IEEE Xplore, ScienceDirect, SpringerLink, Scopus, Web of Science, and Google Scholar.

2. **Search Strategy and Study Selection**

   (a) Search strings were constructed using keywords and Boolean operators (e.g., deep learning AND crop disease AND pest management AND decision support).

   (b) These search strings were applied to the selected digital libraries.

   (c) Titles and abstracts were screened to remove duplicates and irrelevant studies.

   (d) Inclusion and exclusion criteria were applied to select relevant full-text articles:

   - Inclusion: Studies focusing on deep learning for crop disease or pest detection with decision support outcomes.
   - Exclusion: Studies that did not involve deep learning models, were not crop-related, or lacked empirical evidence.

3. **Quality Assessment**

   (a) A quality assessment checklist was developed, evaluating clarity of objectives, methodological accuracy, dataset availability, and reproducibility.

   (b) Each study was scored using the checklist to ensure high-quality and reliable findings.

   (c) Only studies that met the minimum quality threshold were included in the final synthesis.

4. **Data Extraction and Coding**

   (a) A data extraction form was designed to collect key attributes such as author, year, crop type, deep learning model used, dataset, decision support approach, and influencing factors.

   (b) Data were systematically extracted and coded into predefined categories.

(c) Extracted data were organized in a tabular format to facilitate analysis.

5. **Synthesis and Thematic Analysis**

(a) A qualitative synthesis was performed to identify recurring themes and patterns.

(b) Influencing factors such as model architecture, data quality, explainability, user integration, scalability, and environmental conditions were analyzed.

(c) Relationships between identified factors and their impact on decision support effectiveness were mapped.

(d) Results were grouped based on the types of deep learning models used (e.g., CNN, RNN, GANs, and Hybrid models).

6. **Validation and Reporting**

(a) Findings were validated through peer review and expert feedback.

(b) Synthesized results were compared with existing frameworks or benchmark studies.

(c) Findings were reported in alignment with PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) guidelines.

(d) Limitations were discussed, and future research directions were suggested, focusing on unresolved factors and emerging technologies.

7. **Output Final Review**

(a) A comprehensive report was produced, outlining the key influencing factors.

(b) Visualizations such as bar charts, thematic maps, and factor relationship diagrams were presented.

(c) Critical insights were highlighted to inform the design and application of deep learning-based decision support systems in agriculture.

Figure 3.4 Systematic Literature Review on Deep Learning Model Factors Influencing Decision Support in Crop Diseases and Pest Management

The purpose of the systematic literature review was to find and evaluate deep learning model factors that affect decision support for crop disease and pest management, as shown in Figure 3.4. In the beginning, the researcher used the Population, Intervention, Comparison, Outcome, and Context (PICOC) framework to precisely describe the study objectives and questions during the systematic review planning phase. The study's scope was defined by the PICOC framework, which emphasized crop leaf diseases and pests as a Population, using deep learning models as an Intervention to manage crop diseases and pests, comparing these models to traditional methods(Comparison), and ultimately, this improved the accuracy of decision-making (Outcome) within agricultural settings (Context). Inclusion and exclusion criteria,

search strategies, and quality evaluation processes were all outlined in the review protocol that was used to guide the approach. For this literature review, the researcher used digital libraries including Google Scholar, Scopus, Web of Science, ScienceDirect, IEEE Xplore, and SpringerLink [209].

Decision support, pest management, deep learning, and crop disease were among the many keywords and Boolean operators used in the exhaustive search technique. The papers that were initially considered were first screened for titles and abstracts, and then their entire texts were reviewed according to predetermined qualifying criteria. Excluded studies lacked empirical data or did not focus on Deep Learning models linked to agriculture; studies that focused on deep learning applications in decision support for crop disease and pest classification were included [210].

The researcher used a uniform criterion to evaluate the research's quality and made sure that only the most accurate and trustworthy ones were chosen. Information such as publication year, crop type, Deep Learning model utilized, dataset specifics, decision support strategies, and affecting variables was gathered via a coding template, which was then used to systematically extract data. For analysis, the retrieved data were tabulated and placed into categories [211].

Considerations, including model design, data quality, interpretability, scalability, environmental adaptation, and user integration, were analyzed using a theme analysis to find commonalities and recurrent aspects. Decision support systems' efficacy was examined in relation to these aspects. The results were confirmed by comparing them to preexisting standards and frameworks and undergoing peer review. For accountability and replicability, the review adhered to PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) standards [212].

As the last step, the results were organized into a detailed report that outlined the most important findings, discussed the factors that had an impact, and made recommendations on how to improve crop diseases and pest management through the use of deep learning models.

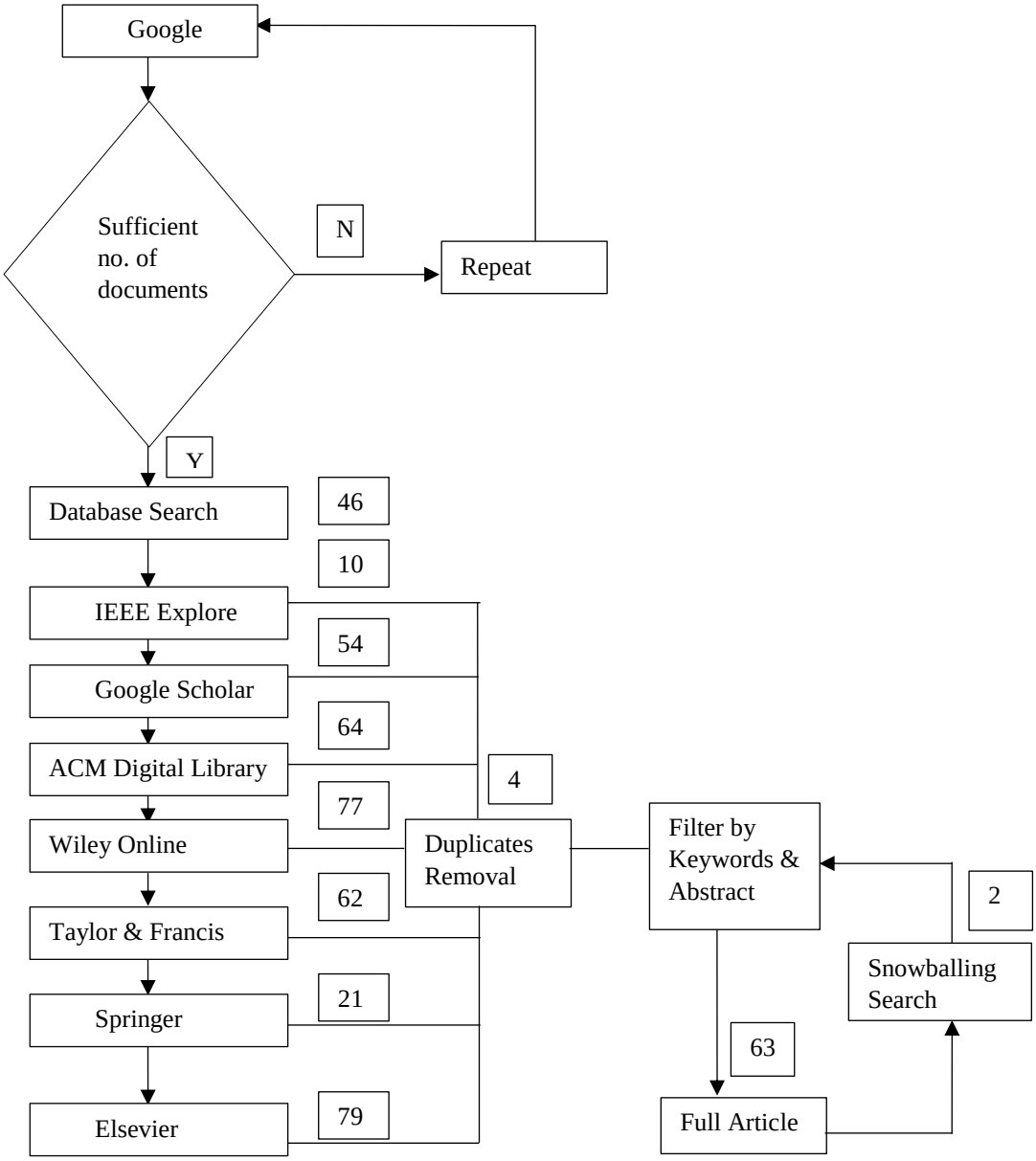


Figure 3.5 Article Selection Process

The systematic literature review process, illustrated in Figure 3.5, began with an initial broad search using Google to identify preliminary documents. The researcher then assessed whether a sufficient number of relevant studies had been retrieved. Since the initial result was negative, the search process was repeated, indicating an iterative refinement of the search strategy. Upon identifying an adequate number of documents, the review progressed to a more structured database search across several academic platforms, resulting in a total of 466 documents. Specifically, 109 were sourced from IEEE Xplore, 54 from Google Scholar, 64 from the ACM Digital Library, 77 from the Wiley Online Library, 62 from Taylor & Francis, 21 from Springer, and 79 from

Elsevier. Following this, duplicate records were removed, reducing the total to 405 documents. These were then filtered based on keywords and abstract relevance, narrowing the selection to 63 articles. Full texts of these articles were retrieved for an in-depth review. Subsequently, a snowballing technique was employed, whereby reference lists of the 63 full-text articles were reviewed to identify additional relevant studies. This yielded two more documents, increasing the final set to 65 articles. The entire review process was systematically designed to identify and analyze the key factors influencing decision support in crop disease and pest management through the application of deep learning models, as demonstrated by the comprehensive and iterative search and selection methodology.

Along with the systematic literature review, experiments were conducted using EfficientNet, CNN, AlexNet, and VGG, the four deep learning models. Using an experimental study approach, the researcher aimed to discover and analyze the primary features that impact decision support in crop disease and pest management using deep learning models. Using this method, the researcher was able to evaluate and compare deep learning models on a range of crop diseases and pests datasets, allowing the researcher to discover how different factors affect the accuracy and consistency of these models in practice. The entire research was carried out using Google Colab, an online platform for deep learning experiments that is both powerful and user-friendly, and which supports Graphics Processing Unit (GPU) acceleration. Training, testing, and evaluating models efficiently across crop diseases and pest datasets was made possible by this architecture, which also allowed for scalability and reproducibility.

The study made use of three public image datasets that depicted various aspects of pest infestation and crop diseases. The first set of data consisted of images of maize leaf diseases, which consisted of leaf spot, common rust, northern leaf blight, and healthy leaves. The second batch of data came from the same place—the PlantVillage repository—and it included both unhealthy and diseased tomato plants, such as early blight, late blight, leaf mold, and healthy samples. Images of pests taken from public domain sources, such as Kaggle, made up the third dataset. Agricultural pests such as aphids, whiteflies, and fall armyworms were included in this collection. These data

sets were selected because of their diversity, ease of use, and relevance to real agricultural problems.

All images were scaled to 224 by 224 pixels before being fed into the models. This was done to make sure they fit the input criteria of conventional Convolutional Neural Networks. To make sure the training was consistent, the pixel values were normalized to a scale of 0 to 1. A number of augmentation methods were used, including random rotations, flips, zooming, and shear transformations, to enhance the training data and reduce overfitting. It was because of this that the models' generalizability was improved. After processing, the datasets were divided into three parts: training, validation, and testing. The ratio of the split data was 70:15:15. All three subsets maintained a balanced class distribution since stratified sampling was used to achieve that.

In order to facilitate comparison, many models were put into place. First, a convolutional, pooling, and fully connected dense unit layer was used to construct a baseline Convolutional Neural Network (CNN). Then, a softmax layer was added for classification. Along with this custom model, the research investigated the possibility of transfer learning using pre-trained networks, including Visual Geometry Group, DenseNet, EfficientNet, and AlexNet. Models were trained using ImageNet weights and then fine-tuned for each classification task by adjusting their top layers. By combining max pooling, dropout layers, and a new softmax output layer proportional to the number of classes in each dataset, the original classifier was removed and replaced with a Support Vector Machine. The CNN was used as a feature extractor in the experiment that used the hybrid model, while a Support Vector Machine (SVM) with a Radial Basis Function kernel handled the final classification. In this scenario, the CNN's penultimate layer was used to extract features, which were then passed to the scikit-learn-based Support Vector Machine classifier.

All models were trained using the same setup. The researcher trained for 20 epochs using the Adam optimizer with categorical cross-entropy loss with a batch size of 32. The researcher utilized validation loss to trigger early stopping and model checkpoints to hold on to the best-performing weights in order to prevent overfitting. There were two phases to training the CNN-SVM hybrid. The CNN was trained on the image data;

thereafter, feature vectors were retrieved and passed to the Support Vector Machine for classification.

Various assessment measures were used to determine how well each model performed. The model's ability to differentiate between classes was assessed using a variety of metrics, such as recall, accuracy, precision, F1-score, and confusion matrix analysis. To further track the model's actions as it learned, the researcher also generated training and validation curves. Several ablation experiments were conducted to delve further into the question of which components significantly affected the performance of the model. This required tuning the configuration in small increments, disabling data augmentation, converting images to grayscale, reducing the dataset size, modifying the usage of dropout layers, and adjusting image resolution. In order to determine which variables substantially influenced the model's outputs, the researcher measured the effect of each adjustment.

Utilizing Grad-CAM (Gradient-weighted Class Activation Mapping), visual explanations were generated to aid in understanding the model predictions. The model's predictions were shown by these heatmaps, which showed which parts of each image the model focused on while making predictions. This enabled the researcher to confirm that the model was focusing on areas impacted by pests or crop diseases and not random noise or irrelevant attributes.

This methodology offered a detailed foundation for understanding the effectiveness of various deep learning models when applied to the problem of pest and crop disease control in agricultural production. A comprehensive investigation of the factors influencing the classification, reliability, and utilization of deep learning models in agricultural decision support systems was made possible by using a wide variety of crop diseases and pests datasets, different types of models, systematic metrics for assessment, and visual aids for interpretation [213].

3.10 Development of an Enhanced Deep Neural Network Model

This research used secondary datasets obtained from Kaggle to develop an enhanced deep neural network model that could reliably identify and classify pests and crop diseases for decision support. There were three primary types of images included in the datasets: pests, maize leaf diseases, and tomato leaf diseases. Among the 7,316

images in the maize dataset were four classes: Common Rust, Healthy Leaves, Leaf Spot, and Northern Leaf Blight. There were a total of 10,000 images in the tomato dataset, which was divided into 10 classes, indicating different sorts of diseases, including Early Blight, Late Blight, Leaf Mold, and Septoria Leaf Spot. Additionally, there was a healthy class. Corn weevils, fall armyworms, and aphids were among the 12 pest classes included in the 4,395 pest images dataset.

Google Colab and the PyTorch deep learning framework were used to create the experimental setup. The researcher utilized three datasets consisting of maize, tomato, and pest images. Of the three datasets, 7,316 images were from maize, 10,000 from tomatoes, and 4,395 from pests. Before training, every image was scaled to 224 × 224 pixels to ensure consistency. Every dataset had its own 70% training set, 15% validation set, and 15% testing set.

The training data was enhanced using image augmentation methods such as random rotations (±25°), horizontal and vertical flips, random zoom (0.8-1.2×), brightness and contrast adjustments, and Gaussian noise injection in order to make the model more robust and less prone to overfitting. With an initial learning rate of 0.0001 and a categorical cross-entropy loss function, the model was trained for 20 epochs with a batch size of 32 using the adam optimizer.

The experiments were carried out in the Graphical Processing Unit runtime environment of Google Colab on a 16 GB VRAM, 2.20 GHz Intel Xeon CPU, and 12 GB of processing RAM NVIDIA Tesla T4 GPU. For augmentation, the researcher used PyTorch's torchvision.transforms package. To speed up execution, the researcher set up DataLoader to shuffle training batches and allow parallel data loading [215].

Table 3.4 Dataset Sizes, Augmentation Techniques, and Training Parameters

Dataset	No. of Images	Image Size	Augmentation Techniques	Epochs	Batch Size	Optimizer	Learning Rate
Maize	7,316	224×224	Rotation ±25°, flips (H/V), zoom, brightness/contrast, Gaussian noise	20	32	Adam	0.0001
Tomato	10,000	224×224	Rotation ±25°, flips (H/V), zoom, brightness/contrast, Gaussian noise	20	32	Adam	0.0001
Pest	4,395	224×224	Rotation ±25°, flips (H/V), zoom, brightness/contrast, Gaussian noise	20	32	Adam	0.0001

The datasets, augmentation approaches, and training settings used in the experiment are summarized in Table 3.4. It describes the amount of images in each dataset, which includes 7,316 images for maize, 10,000 images for tomatoes, and 4,395 images for pests. In order to guarantee consistency and quality, all datasets were preprocessed before model training. Image scaling was 224 × 224 pixels, pixel normalization was within a range of 0–1. The following augmentation techniques were used across all datasets: ±25° rotations, horizontal and vertical flips, random zoom, brightness and contrast adjustments, and Gaussian noise to enhance the model's generalizability and make it less prone to overfitting. Also included are the training parameters, which reveal that the Adam optimizer was used with a learning rate of 0.0001 and a batch size of 32 throughout the 20-epochs training process for each dataset.

A Convolutional Neural Network (CNN) was used for feature extraction in the development of the hybrid deep learning model. A Support Vector Machine (SVM) classifier was used for final classification. Convolutional Neural Network (CNN) layers used the images to extract high-level spatial features, which the Support Vector Machine (SVM) used to classify the images into their respective classes [214].

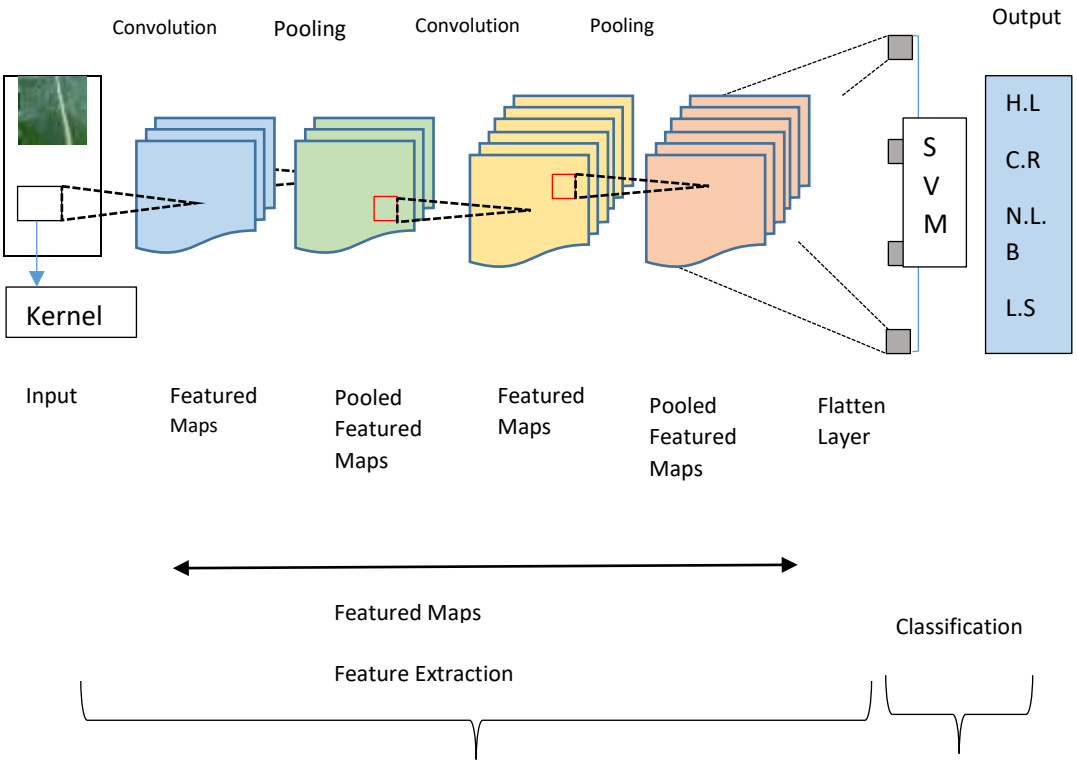


Figure 3.6 CNN-SVM Model

Considering image A and kernel B, both of them being 2 matrices, and in the field of computer vision, the images were expressed in terms of pixels. The output was produced by matrix C when the researcher convolved image A with kernel B.

1	2	3
2	0	0
7	9	1

*

3	2	0
3	0	1
0	5	2

=

1*3 = 3	2*2 = 4	3*0 = 0
2*3 = 6	0*0 = 0	0*1 = 0
7*0 = 0	9*5 = 45	1*2 = 2

The summation of all the elements in matrix C was done to get a scalar number i.e. 3 + 6 + 0 + 4 + 0 + 45 + 0 + 0 + 2 = 60.

The image features included colour, texture, diagonal, arch, horizontal edges, vertical edges, and ridges. The kernel enabled one to measure how close an input matches the features. The similarities of different regions in the input image were found by convolving the kernel across the image [216].

3.10.1 Convolution Neural Network

The arrangement of convolution layers and max pooling layers produced good results based on a particular use case. The image classification problem was solved accurately by dividing the images into smaller parts by converting them to a low-dimensional space without losing the information, which in turn reduced the computational complexity. The images were represented in the form of a pixel, and the array of the pixel was in the form of an n x 1 matrix. The resultant matrix was then connected to the hidden layer. Each connection had n weights that were estimated using the backpropagation. Therefore, that means an estimation of n weights needed to be done to connect to the hidden layer. The first hidden layer had more than one node, and each additional node added n weights that needed to be estimated. However, another problem was that it was not clear whether the neural network still performed well if the image was shifted by one pixel. Even complex images tend to have correlated pixels; for example, a brownish pixel in the image was close to other brown pixels. And white pixel tends to be near other white pixels, and it was helpful since the researcher took advantage of the correlation that existed among each pixel. Thus, Convolutional Neural Network helped in the classification of large and complex images. To solve the image classification problem better, the following were considered;

i. The number of input nodes was reduced from 1,024 to 512 to streamline processing.
ii. Minor variations of up to 3 pixels in the positioning of image elements were tolerated.
iii. Identified patterns and relationships in complex images, such as recurring textures and colour gradients, were leveraged to enhance model accuracy.

Applying a Filter (Kernel) to an Input Image

A filter is a small square that is 3 pixels by 3 pixels and finds the similarities between the input images when convolved across it. Hence, the researcher used a 3 x 3 convolution due to the shape of the kernel. The intensity of each pixel in the filter was determined by backpropagation. In other words, before training a Convolutional Neural Network, the researcher started with random pixel values, and after training with backpropagation, the operation ended up with something more useful. To apply the filter to the input image, the researcher overlaid the filter onto the image and then multiplied each overlapping pixel, and then added each product together to get a final value. And the sum of products was called the dot product [217].

Input Image

0	0	1	1	0	0
0	1	0	0	1	0
1	0	0	0	0	1
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0

Kernel

0	0	1
0	1	0
1	0	0

In the computation of the dot product between the input and the filter, it can be said that the filter was convolved with the input. The bias term was then added to the output of the filter, and the final value was placed into something known as the feature map. The process was repeated several times until the feature map was filled up. Since each

cell in the feature map corresponds to a group of neighbouring pixels, the feature map helped take advantage of any correlations that were in the image [218].

Non-linearity

The non-linearity component in the network makes it more powerful, hence improving its performance. This was achieved by making the activation function of the network known as the ReLU function to receive the weighted sum of the input data. The Rectified Linear Unit (ReLU) was used as an activation function in the Convolutional Neural Network (CNN). It was a non-linear function that introduced non-linearity into the output of the CNN. The ReLU function applied an element-wise operation to the output of a convolutional layer, taking the maximum value between 0 and the output value. Mathematically, the ReLU function was defined as:

$$f(x) = \max(0, x) \qquad \text{(Eq. 3.1)}$$

where x was the output of the convolutional layer, and f(x) was the output of the ReLU function.

The ReLU function has several advantages over other activation functions, such as the sigmoid and hyperbolic tangent functions. One of the main advantages was that it was computationally efficient, as it simply involved comparing values to 0. The feature map was then run through a ReLU activation function. This means that all of the negative values were set to 0, and the positive values remained the same as before. The final feature map was the result of the ReLU function applied to it and not the actual sum. The network achieved its true potential when the convolution involved the ReLU operation [219].

Stride and Padding

The filter was moved at each step during the convolution operation by a specified stride of the value 1. Less overlap between the receptive fields was experienced when bigger strides were used. The feature map was reduced whenever the stride got bigger, hence skipping over potential locations. The convolution was contained in the input; hence, the input was larger than the size of the filter. The input was surrounded with zeros through a concept known as padding to maintain the same dimensionality. The input matched the dimensionality of the feature map through padding. To prevent the

size of the feature map from shrinking at each layer, padding was used as a mechanism to avoid that in the Convolutional Neural Network [220].

The formula for the stride in a CNN was as follows:

$$\text{Output size} = ((\text{Input size} - \text{Filter size}) / \text{Stride}) + 1 \quad \text{(Eq. 3.2)}$$

Here, Input size refers to the size of the input feature map, Filter size refers to the size of the filter, and Stride refers to the stride.

For example, an input feature map I of size 5 x 5 x 3 and a filter of size 3 x 3 x 3. The researcher applied convolution with a stride of 2; the output feature map O was of size:

((5 - 3) / 2) + 1 = 2

Therefore, the output feature map O was of size 2 x 2 x D, where D was the depth (i.e., the number of channels) of the filter.

During convolution with a stride of 2, the filter was moved 2 pixels (or units) at a time in both the horizontal and vertical directions over the input feature map. This resulted in the output feature map being down-sampled by a factor of 2 in both dimensions.
The formula for the output size of the feature map with padding was as follows:

$$\text{Output size} = ((\text{Input size} - \text{Filter size} + 2*\text{Padding}) / \text{Stride}) + 1 \quad \text{(Eq. 3.3)}$$

Here, Input size refers to the size of the input feature map, Filter size refers to the size of the filter, Padding refers to the number of border pixels added to the input feature map, and Stride refers to the stride of the convolution operation.

Padding was usually denoted by P. P was the number of pixels added to each side of the input feature map, then the total amount of padding was 2P. There are two types of padding: valid and same.

Valid padding: No padding was added to the input feature map, so P = 0. The output feature map had smaller spatial dimensions than the input feature map.

Same padding: Padding was added to the input feature map so that the output feature map had the same spatial dimensions as the input feature map. In this case, the amount of padding was given by;

$$P = ((\text{Stride} - 1) * (\text{Input size} - 1) - \text{Input size} + \text{Filter size}) / 2 \quad \text{(Eq. 3.4)}$$

For example, an input feature map I of size 5 x 5 x 3 and a filter of size 3 x 3 x 3. The researcher applied convolution with a stride of 1 and the same padding. The amount of padding required was:

P = ((1 - 1) * (5 - 1) - 5 + 3) / 2 = 1

Therefore, the researcher added 1 pixel of padding to each side of the input feature map to ensure that the output feature map had the same spatial dimensions as the input feature map.

The resulting padded input feature map I' was of size 7 x 7 x 3.

Pooling

The dimensionality reduction was done after the convolution operation through pooling. This technique combated the overfitting problem and reduced training time since it reduced the number of parameters. The process kept the depth of the feature map constant and reduced the height and width since the pooling layer down-sampled each feature map. The pooling had no parameters, contrary to the convolution operation. The maximum value was selected from the window after sliding the window over the input. The stride and window size were specified similarly to the above convolution operation.

The new feature map was subjected to another filter; however, unlike before, the maximum value was selected. The filter moved in such a way that it did not overlap itself. When the maximum value was selected in each region, the concept was known as max pooling, and it clearly showed that the technique selected the spots where the filter did the best job matching the input image. Alternatively, average or mean pooling was done by calculating the average value for each region. The pooled layer was then converted into a column of input nodes. The algorithm can be summarised as follows;

i. The input image, sized 128×128 pixels, was read.
ii. It was processed through a 3×3 filter to generate a 126×126 feature map.
iii. The feature map was passed through a ReLU activation function to introduce non-linearity.

iv. From each 2×2 region, the maximum value was selected during max pooling, producing a 63×63 pooled feature map.
v. The pooled feature map was flattened into a column vector of 3,969 values for the input nodes.
vi. Each input node value was multiplied by its corresponding weight (ranging from −0.5 to 0.8), the results were summed, and a bias of 0.1 was added.
vii. The resulting value was then passed through the ReLU activation function to obtain the final activation output.

An example of a result for stride 2 and a 2 x 2 window when performing max pooling is shown below. The window was not overlapping since the stride used was 2, together with a 2 x 2 window size, and each colour denotes a different window [221].

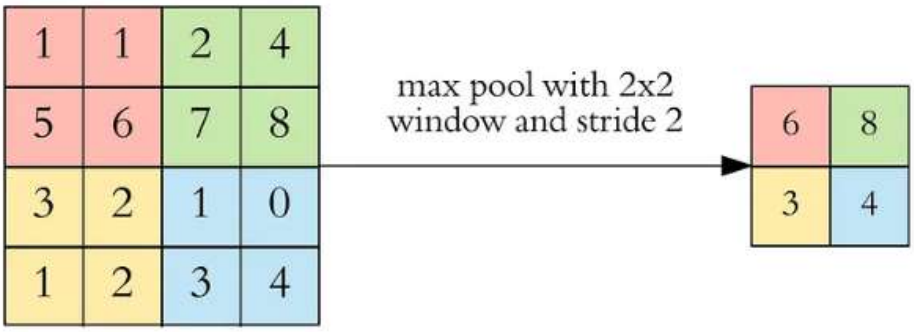


Figure 3.7 Max Pooling

Using an image of 32 x 32 x 10 with the same pooling operation and parameters, the height and width were halved, and the depth remained the same because on each depth slice of the input, the pooling operation worked independently.

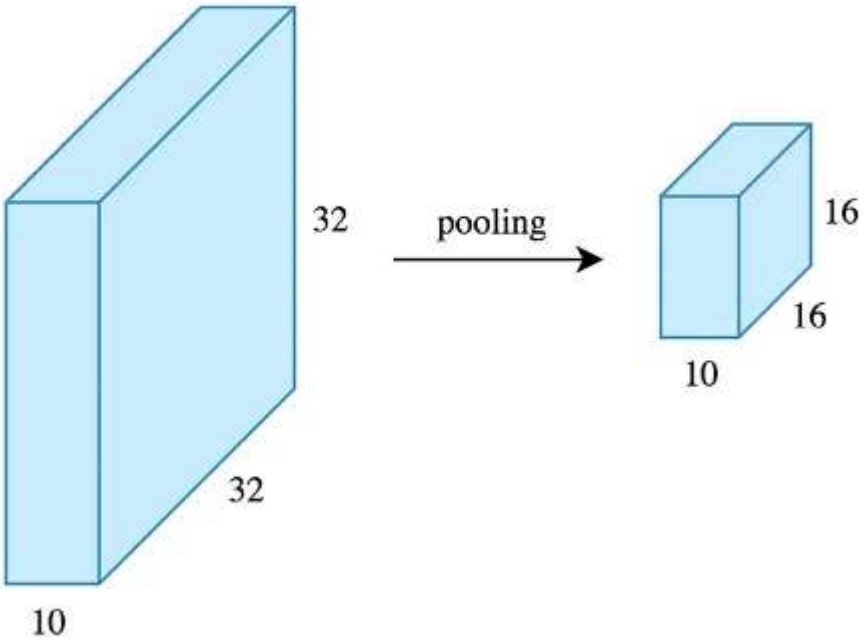


Figure 3.8 Pooling Operation

The weights were reduced to ¼ of the input by halving the height and the width. This was a pretty nice idea since in Convolution Neural Networks we deal with millions of weights, hence weight reduction reduces the training time, and the overfitting problem, hence improving the image classification accuracy.

Here's a step-by-step breakdown of how the max pooling operation worked on a single channel (when the pool size was 2x2 and the stride was 2);

i. The first 2×2 region in the 64×64 input feature map was selected.
ii. The maximum value within this 2×2 region was identified (e.g., if the region contained [3, 5, 2, 7], the maximum value was 7).
iii. The pooling window was shifted 2 pixels to the right (stride = 2), and steps i and ii were repeated until the end of the row was reached, producing 32 values across the first row.
iv. The pooling window was then moved 2 pixels down, and the same process was repeated across all rows until the entire 64×64 feature map was scanned.
v. The resulting output feature map was reduced to 32×32, with each value representing the maximum element from its corresponding 2×2 region in the original input feature map.

The formula for max pooling in a Convolutional Neural Network (CNN) was expressed mathematically as follows;

An input feature map with dimensions W × H × D was used to determine the pooling size as F and the stride as S. The researcher used the formula below to calculate the size of the output feature map;

$$\left(\frac{W-F}{S}+1\right)\times\left(\frac{H-F}{S}+1\right)\times D \qquad \textbf{(}\text{Eq. 3.5)}$$

The output feature map was reduced to 32 × 32 × 32 from 64 × 64 × 32 in the input feature map, where the pooling size was F=2F, and the stride was S=2S.

The input feature map was divided into non-overlapping regions of size F×F for each channel i. For channel i in the output feature map, the largest value within each region was chosen, and in the feature map, each channel was split into 2×2 sections. For example, given an input feature map that was 64×64×32 in size and a pooling size of 2 with F=2. A feature map with dimensions of 32 × 32 × 32 was generated by extracting the maximum value from each region.

The output feature map O at position (x, y, i) was computed as follows;

$$O(x, y, i) = \max(I(x', y', i)) \text{ for all } x' \text{ in } [xS, xS+F-1], y' \text{ in } [yS, yS+F-1] \qquad \textbf{(}\text{Eq. 3.6)}$$

Where I was the input feature map, and the max() function computed the maximum value within the pooling window.

The operation retained the important features, hence making computation more efficient through a reduction in the spatial dimension of the feature map. The complex features were still extracted by the downstream layer despite the output feature map being smaller than the input feature map. The high-level representation of the feature map or the input image and the abstract features were captured by the network when the spatial dimension was reduced [222].

Note: The Convolution Neural Network helped reduce the number of inputs into the neural network, and this was seen when the researcher took 6 x 6 images, or 36 potential inputs, which were compressed down to just 4 inputs into the neural network.

The Convolutional Neural Network took correlations into account, and this was accomplished by the filter, which looked at a region of pixels, instead of just one pixel at a time.

The Convolutional Neural Network could handle minor changes in the locations of individual pixels in the image. To improve the interpretability of the output, the researcher passed the results to the softmax function, which converted the raw scores into normalised probability values for each class.

3.10.2 The Softmax Layer

The layer associated with image classification in deep learning was known as the softmax layer. The layer calculated the probability for each class to which the input belonged. The number of softmax units in the output layer was equal to the number of classes, so that each unit could hold the probability of each class. This was also known as a probability distribution where the sum of the probabilities for all the classes was equal to 1. The softmax layer used the following equation to calculate the probability for each class;

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_j^k = 1^{e^{z_j}}} \quad \text{(Eq. 3.7)}$$

σ=softmax

$\vec{z}$=input vector

e^{z_i} =standard exponential function for input vector

K = number of classes in the multi-class classifier

e^{z_j} =standard exponential function for output vector

To ensure the sum of all exponentiated values adds up to 1, each element of the output layer was subjected to a standard exponential function and then divided by the sum of all exponentials to normalize [223].

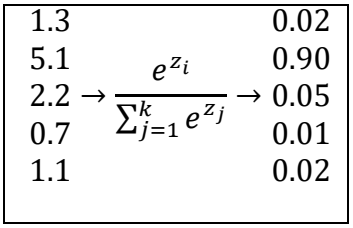


Figure 3.9 Softmax Layer

Steps

i. The exponent of each element in the output layer was calculated, and the results were summed, giving a total of 181.73.
ii. Each exponent value was then divided by this sum. For example, for an output value of 1.3, the exponent was $e^{1.3} = 3.67$. Dividing this by 181.73 yielded a probability of 0.02.

The predicted class was the one with the highest confidence score in the probability distribution list.

All the probability distribution values were converted to zero except the highest probability, which was converted to 1 by use of the Argmax function using the equation below;

$$\arg\max\left(\begin{bmatrix} z_1 \\ z_2 \\ .. \\ z_n \end{bmatrix}\right) = \begin{bmatrix} y_1 \\ y_2 \\ .. \\ y_n \end{bmatrix} = \begin{bmatrix} 0 \\ .. \\ 1 \\ .. \\ 0 \end{bmatrix} \quad \text{(Eq. 3.8)}$$

where $y_i = 1$ iff z_i is the unique maximum value of $\vec{z}$

It was good practice to infer the model using the argmax layer since it produced a single value rather than a probability, and the researcher trained the classifier while using the softmax layer since it allowed the researcher to optimise the cost function [224].

3.10.3 The Support Vector Machine

In this study, a hybrid approach was used where the Convolutional Neural Network (CNN) was used to extract image features, which were then input into a Support Vector Machine (SVM) for the purpose of classification. While the Support Vector Machine

(SVM) did the actual classification using the extracted features, the Convolutional Neural Network (CNN) learnt a representation of the input data. By reducing the generalisation error on unseen data, the Support Vector Machine improved the classification accuracy. Since SVM scaled very well to high-dimensional data, it outperformed the softmax layer when it came to generalising the training dataset. In order for a Support Vector Machine (SVM) to correctly classify the images, the ideal plane, also called the hyperplane, was a line that divided the data into their respective classes.

Where w was the vector that was typically normal to the hyperplane and b was an offset, the hyperplane equation was written as $w.x + b = 0$.

In order to use the hyperplane to separate the data points, the following steps were followed:

i. A reference line and two parallel lines, placed at equal distances apart, were drawn.
ii. A large number of repetitions (epochs) was selected, with the value set to 1000.
iii. An expansion factor close to 1 was used, for instance, 0.99.
iv. The process was repeated a thousand times.
    For each iteration, an arbitrary point was selected.
    If the classification of the point was correct, no adjustment was made.
    However, if the point was misclassified, the separating line was shifted closer to that point.
    The expansion factor was then applied to adjust the spacing of the lines.
v. The final separating lines were correctly positioned to distinguish the data points.

Since the following equation maximised the margin of the hyperplane and the data points, a larger margin around the decision boundary improved the classification accuracy.

$$c(x, y, f(x)) = \begin{cases} 1 - y * f(x), \text{if } y.f(x) \geq 1 \\ 0, \qquad \text{else} \end{cases} \qquad \text{(Eq. 3.9)}$$

To help maximise the margin between the hyperplane and the data points, the above equation, called the hinge loss function, was used. When the cost function was 0, the sign of the actual value and the anticipated value were identical. The loss value was determined if they differ. To get the cost function, the researcher took the loss function

and averaged it. Gradient Descent was an optimisation technique that was used to minimise the cost function. Using the optimisation process, the learning model's parameters were adjusted by determining the derivatives with respect to the weights, and the gradient was computed from the loss function [225].

$$w_2 = w_1 - L * \frac{d_j}{d_w} \quad \text{(Eq. 3.10)}$$

$$b_2 = b_1 - L * \frac{d_j}{d_b} \quad \text{(Eq. 3.11)}$$

$w \rightarrow$ weight

$b \rightarrow$ bias

$L \rightarrow$ Learning Rate

$\frac{d_j}{d_w} \rightarrow$ Partial derivative of the cost function with respect to w

$\frac{d_j}{d_b} \rightarrow$ Partial derivative of the cost function with respect to b

Note:

$\frac{d_j}{d_w} \rightarrow$ was how much the cost function changed when the weight changes.

$\frac{d_j}{d_b} \rightarrow$ was how much the cost function changed when the bias changes.

$\frac{d_j}{d_w}$ and $\frac{d_j}{d_b}$ were known as gradient or derivatives.

The margin maximisation and the loss were balanced by adding the regularisation parameter to the cost function, which was as follows;

If $(y_i \cdot (w \cdot x+b) \geq 1$:

$$\frac{d_j}{d_b} = 2\lambda w$$

$$\frac{d_j}{d_b} = 0$$

else $(y_i \cdot (w \cdot x+b) < 1$:

$$\frac{d_j}{d_w} = 2\lambda w - y_i . x_i$$

$$\frac{d_j}{d_b} = y_i$$

When there was zero misclassification, the model correctly assigned a class to each of the data points. Here, the gradient was updated using the regularisation parameter.

$$w = w - \alpha.(2\lambda w) \quad \text{(Eq. 3.12)}$$

When the developed model got the image classifications wrong, the regularisation parameter and the loss function were both used to determine the gradient update.

$$w = w + \alpha.(y_i . x_i - 2\lambda w) \quad \text{(Eq. 3.13)}$$

The difference between the two classification modules was that the softmax layer maximised the log-likelihood or minimised the cross-entropy, and the Support Vector Machine determined the maximum distance between support vectors of different classes.

In this research, the combination of a Convolution Neural Network and Support Vector Machine was implemented by the following architecture;

INPUT:256 x 256 x1

(conv_layers): Sequential(

(0): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))

(1): ReLU()

(2): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

(3): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))

(4): ReLU()

(5): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

(6): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)

(7): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))

(8): ReLU()

(9): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

(10): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))

(11): ReLU()

(12): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

(13): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
(14): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))

(15): ReLU()

(16): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

(17): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))

(18): ReLU()

(19): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

(20): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
(21): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))

(22): ReLU()

(23): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

(24): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))

(25): ReLU()

(26): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

(27): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)

(dense_layers): Sequential(

(0): Dropout(p=0.5, inplace=False)

(1): Linear(in_features=50176, out_features=1024, bias=True)

(2): ReLU()

(3): Dropout(p=0.4, inplace=False)

(4): Linear(in_features=1024, out_features=4, bias=True)

def forward(self, output):

out = self.conv_layers(output)

out = out.view(-1, 224*224)

out = self.dense_layers(out)

Note: The weight parameters were learned by using Adam instead of the softmax layer in the Convolutional Neural Network, and the Support Vector Machine was used for the classification problem.

Algorithm: Hybrid CNN-SVM Model for Crop Diseases and Pest Classification

Input: Image dataset of crops with labeled disease and pest classes

Output: Trained hybrid CNN-SVM model for classification

Procedure:

Data Collection and Preprocessing

(a) A dataset of crop images labeled with various diseases and pests was collected.

(b) The images were preprocessed by resizing, normalizing, and applying data augmentation to increase variability.

CNN Model Training

(a) A CNN architecture with multiple convolutional and pooling layers was initialized.

(b) The CNN was trained on the preprocessed dataset.

(c) Parameters such as the number of layers, filter sizes, and learning rate were optimized to maximize accuracy.

(d) Features were extracted from the final fully connected layer of the CNN.

SVM Classifier Training

(a) The CNN-extracted features were used as input to the SVM classifier.

(b) An RBF kernel was selected, and its parameters (gamma and C) were tuned through cross-validation.

(c) The SVM was trained to classify the extracted features into the respective crop disease and pest classes.

Evaluation and Comparison

(a) The hybrid CNN-SVM model was evaluated using the test set.

(b) Performance metrics including accuracy, precision, recall, and F1-score were measured.

(c) The results were compared against those of the standalone CNN and other baseline models such as DenseNet and VGG for validation.

Final Output

The final hybrid CNN-SVM model was obtained for crop disease and pest classification.

Figure 3.10 CNN-SVM Algorithm

To develop the hybrid CNN-SVM model for crop disease and pest classification, data had to be collected and preprocessed. The range of real-world crop variations was captured by first collecting a diverse image collection from the Kaggle website, which included labels for crop diseases and pests. The researcher ensured that all images

were consistent by scaling and normalising them, and then used data augmentation techniques such as flipping, rotating, and colour alteration to boost model generalizability and introduce variability.

After that, the Convolutional Neural Network (CNN) model was trained using pooling and convolutional layers to detect specific crop diseases or pests using their textures and patterns in the images. The preprocessed images were fed into the CNN network to train it to learn these features quickly. The researcher tuned the learning rate, filter size, and number of layers, among other crucial factors, to make the model more accurate. The last fully connected layer of the CNN after training yielded feature extractions, which retained crucial visual information [226].

The features were then used to train a Support Vector Machine classifier. Because of its exceptional performance in handling high-dimensional data and in determining optimal decision boundaries, the SVM was chosen. The Radial Basis Function kernel was chosen by the researcher because of its effectiveness in solving nonlinear classification issues. To get optimal results, cross-validation was used to fine-tune parameters such as gamma and C [46].

The model's performance was evaluated using a test dataset. Using metrics such as F1-score, recall, accuracy, and precision, the researcher examined the effectiveness of the classification process. Verification of the hybrid model's efficacy was aided by experiments that compared it against DenseNet, VGG, and standalone CNN models, among others. The final product was a trained CNN-SVM model that improved classification performance for crop diseases and pests, suggesting it should be deployed in the agricultural environment [227].

3.11 Software Platform Implementation and Integration

A web-based system was created, tested, and put into practice to incorporate the developed deep learning model into an intuitive software platform that offered prompt and practical decision-making support. The first step in implementing the system was choosing the right technology to meet the functional and performance needs of the system. Users were able to effortlessly upload crop images, obtain classification results, and access pertinent information through the platform's frontend, which was built using HTML5, CSS3, and JavaScript to provide a dynamic, responsive, and

aesthetically appealing experience. To make it more accessible and usable on a variety of devices and screen sizes, the researcher utilised the Bootstrap framework and added some custom styles [228].

To manage data flow between components, interface with the trained hybrid CNN and SVM model, and handle user requests, the backend relied on FastAPI, a high-performance Python web framework. Using Python tools like scikit-learn and TensorFlow, the trained and tested model was serialised and fed into the backend with labelled images of maize, tomato leaf disease, and pests. The backend received the image data from the frontend over RESTful APIs, processed it, and then sent it to the model for classification. Classification confidence scores and predicted class labels (such as healthy, common rust, northern leaf blight, leaf spot, or pest) were both returned by the model. The backend system used the detected class to get pest or crop disease management support, including the necessary fungicides and insecticides, from a pre-defined database [229].

Users were able to find the closest agrovets depending on their geolocation due to the platform's integration with the Google Maps API, which enhanced decision support. Users were able to access the agrovet shops' contact information, physical locations, and navigation options. A database layer was also a part of the system; it used SQLite to keep track of agrovet data, user queries, and classification logs. The implementation of secure image upload techniques and the explicit permission to retain user data beyond the session interval were measures taken to address security and privacy concerns [230].

To ensure the software platform was accurate, reliable, and resilient, it went through iterative testing stages, including User Acceptance Testing (UAT), unit testing, and integration testing. Farmers and agricultural officials made up the sample users who participated in the field trials that assessed the system's practicality and ease of use. The results of these tests helped the researcher fine-tune the interface and boost the system's efficiency. By providing users with practical, easily accessible, and accurate agricultural insights in real time, the platform effectively connected the Artificial Intelligence-based classification model with end-users [231].

Algorithm: Implementation of a Deep Learning Model in a Web-Based Decision Support System

Input: User-uploaded image of a crop leaf disease or pest

Output: Classified disease/pest, confidence level, recommended control measures, nearest agrovet location

Step 1: System Initialization

1.1 The trained hybrid CNN-SVM model was loaded from the server storage.

1.2 The FastAPI backend application was initialized.

1.3 A connection was established to the local SQLite database containing pesticide/fungicide and agrovet information.

1.4 The frontend interface was designed using HTML, CSS, and JavaScript.

1.5 The Google Maps API was integrated for location services.

Step 2: User Image Upload (Frontend)

2.1 The user accessed the web interface via a browser.

2.2 A crop image was uploaded through the image upload form.

2.3 The uploaded image was sent to the FastAPI backend via a POST request.

Step 3: Image Preprocessing (Backend)

3.1 The image file was received on the server.

3.2 The image was resized and normalized to match the model input dimensions.

3.3 The image was converted into a compatible array format.

3.4 Input features were prepared for the hybrid CNN-SVM pipeline.

Step 4: Image Classification

4.1 The preprocessed image was passed into the CNN to extract deep features.

4.2 The extracted features were passed to the SVM for final classification.

4.3 The output class (e.g., Common Rust) and prediction confidence score (e.g., 96.9%) were captured.

Step 5: Decision Support Retrieval

5.1 The database was queried for the recommended pesticide or fungicide corresponding to the classified disease/pest.

5.2 The system response was formatted to include:

- Crop disease/pest name
- Confidence score
- Recommended treatment
- Preventive measures (if available)

Step 6: Agrovet Location Assistance

6.1 The user's location was retrieved via the browser or manually entered address.

6.2 The Google Maps API was queried to identify the nearest agrovets within a specified radius.

6.3 Details such as name, address, contact number, and distance were extracted.

6.4 A directions link to each agrovet was included.

Step 7: Response Delivery (Frontend)

7.1 Classification results were displayed on the user interface.

7.2 The recommended pesticide/fungicide with usage instructions was shown.

7.3 The nearest agrovets with clickable maps and contact information were listed.

7.4 An option to upload another image or provide feedback was included.

Step 8: Logging and Feedback

8.1 Image metadata and classification results were stored in the database for audit and model improvement.

8.2 User feedback (if provided) was recorded to enhance usability.

8.3 The session ended, and the system returned to Step 2 for another image upload.

End of Algorithm

Figure 3.11 Implementation of a Deep Learning Model in a Web-Based Decision Support System

Figure 3.11 illustrates the detailed process that was followed to incorporate the deep learning model that was created into a web-based decision support system that was developed to classify crop diseases and pests. Loading the trained hybrid CNN-SVM model was part of the initialisation process, as was setting up the FastAPI backend, creating the frontend interface using web technologies, and connecting to both a local database and the Google Maps API. The platform's backend server underwent preprocessing whenever a user uploaded an image depicting a crop disease or pest. The image had to be resized, normalised, and converted to a format that the deep learning model could understand [232].

After the image was preprocessed, it was fed into the Convolution Neural Network to extract deep features. The Support Vector Machine component then used these features to classify the images and identify the particular crop disease or pest. Along with the anticipated class, the algorithm also produced a confidence score as output. The algorithm then found appropriate control recommendations in a predetermined database, such as pesticides or fungicides, depending on the classification result. In

addition, the system displayed directions and contact information for local agrovet shops that were found using the user's geolocation and the Google Maps API. In conclusion, the system's responsive online interface provided the user with all the results, including the classification outcome, the recommended treatment, and information about the closest agrovet. The platform's ability to provide user-friendly, accurate, and quick decision support was guaranteed by this algorithm [233].

3.12 Model Validation

The purpose of model validation was to compare the developed model's performance to that of other pre-existing image classification models. The main reason for doing this was to test how well the developed model performed when subjected to unseen data. By using a confusion matrix, a visual representation was generated that made it easy to see which predicted images were accurate and which ones were incorrect. The Recall, F1 score, accuracy, and precision were all computed with the use of the confusion matrix for the developed model [234].

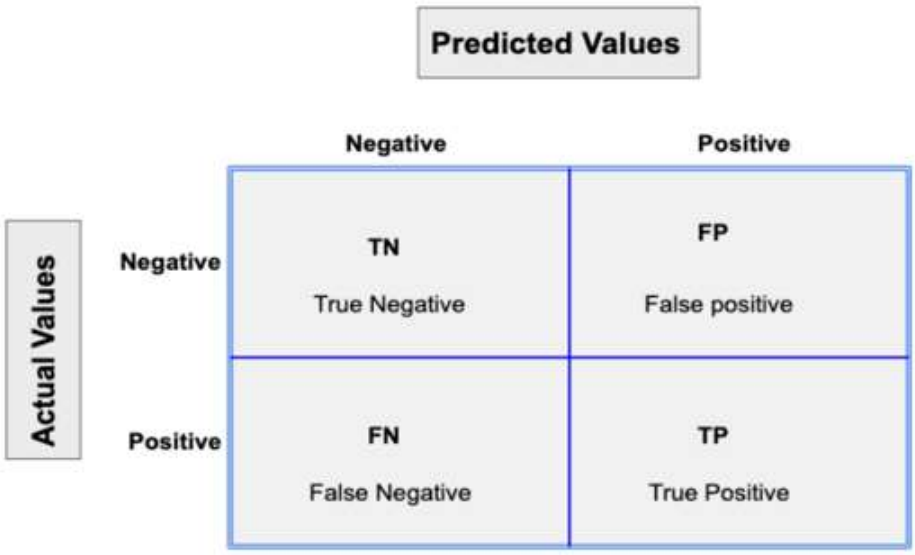


Figure 3.12: Confusion Matrix

Based on the comparison between each prediction and the actual value, the results were categorized as follows;

- **True Positive (TP):** Predicted True and True in reality.
- **True Negative (TN):** Predicted False and False in reality.
- **False Positive (FP):** Predicted True and False in reality.
- **False Negative (FN):** Predicted False and True in reality.

3.12.1 Classification Accuracy

The model's performance was evaluated by its classification accuracy, which was defined as the number of accurate predictions produced by the model.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{(Eq. 3.14)}$$

$$\text{Accuracy} = \frac{\text{Numberof correct predictions}}{\text{Total number of predictions made}} \quad \text{(Eq. 3.15)}$$

3.12.2 Precision

This metric was used to quantify the frequency with which the model correctly classified the image into the correct class.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad \text{(Eq. 3.16)}$$

3.12.3 Recall

It was used to quantify the frequency with which the model correctly predicted yes when the answer was yes.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad \text{(Eq. 3.17)}$$

3.12.4 F1 Score

The result was obtained by determining the harmonic mean of recall and accuracy. F1 score values were between 0 and 1, with 1 being the very best case scenario.

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad \text{(Eq. 3.18)}$$

3.12.5 Training and Validation Loss

When the model was trained, the training loss was the amount by which its predictions deviated from the true labels. It is a vital measurement of the model's progress as it learns the training data. Keeping an eye on training loss when data augmentation methods were used was a good way to make sure the model was properly adjusting to the augmented data and improving generalisation.

On the other hand, the model's performance was evaluated by validation loss, which was based on data that was not utilised during training. One of the most important measures of the model's performance was how well it adapts to new data that it has never seen before. Validation loss was crucial when utilising dimensionality reduction methods to evaluate the feature space reduction's ability to generalise to new data. When validation loss increased, it was because important data required for correct classification was lost during the dimensionality reduction process. A reduction in validation loss, on the other hand, was indicative of successful dimensionality reduction that reduced the computational complexity of the model without diminishing its performance [235].

Overfitting and other problems can only be found by comparing training and validation loss data. When a model fits the training data too well, it is said to be overfitting, and this leads to subpar generalisation. A standard indicator of overfitting was a significant increase in training loss with a relatively stable validation loss. Monitoring these loss indicators helped guarantee that the model benefited from changes like data augmentation and dimensionality reduction, which attempted to solve overfitting by fostering more robust generalisation. With the use of data augmentation and dimensionality reduction, the researcher improved the accuracy of image classification by keeping a close eye on and assessing training and validation loss [236].

In most cases, a loss function was used to calculate training and validation loss measures. Mean Squared Error (MSE) is a popular loss function in deep learning for regression tasks, whereas Cross-Entropy Loss, also known as Log Loss or Negative Log-Likelihood, was used for classification. The classification task had the loss function with the following formula;

Cross-Entropy Loss for the classification task;

Loss in training (measured in terms of the cross-entropy)

$$L_{CE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \quad \text{(Eq. 3.19)}$$

Validation Loss (Cross-Entropy);

$$L_{CE}, \text{validation} = -\frac{1}{M}\sum_{j=1}^{M}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \quad \text{(Eq. 3.20)}$$

Where:

- N was the total number of training samples.
- M was the total number of validation samples.
- y_i was the actual target distribution for the i-th training sample.
- $\hat{y}_i$ was the predicted probability distribution for the i-th training sample.
- y_j was the actual target distribution for the j-th validation sample.
- $\hat{y}_j$ was the predicted probability distribution for the j-th validation sample.

3.13 Ethical Considerations

In order to avoid any form of bias, the researcher followed all ethical guidelines by obtaining a research letter from Murang'a University of Technology and a research approval from NACOSTI. Since NACOSTI states that any research in Kenya should have research ethical clearance, the researcher followed research ethics by getting a research licensing approval from NACOSTI. The researcher also made sure not to gather any data that wasn't directly related to the research objectives at hand, and the researcher focused only on gathering information that was useful in achieving the set research objectives.

The researcher also knew that bias might be present in the dataset as a whole, introduced by variables such as uneven distribution of crop diseases and pests, various crop varieties, different environmental circumstances, and varied image quality standards. To mitigate the effect of these biases, the dataset was preprocessed using techniques such as data augmentation and normalisation. The information was collected from different agricultural regions. The developed hybrid CNN-SVM model was tested for generalisability, fairness, and robustness using a range of performance metrics and compared with current deep learning models.

Additionally, the researcher took into consideration the ecological impact of the computer resources used for training and testing the model. Deep learning models require a lot of processing power and energy consumption. The researcher optimised the training process by employing efficient model designs, reducing extraneous training iterations, and responsibly leveraging existing computing resources. This led to minimising energy use and reducing the effect of the research operations on the environment.

3.14 Summary

This chapter provides an overview of the methodology used to accomplish the study's goals. The first part of the chapter focuses on the research strategy that was chosen, which is a quantitative technique for studying and modelling the intricate interactions in the field of crop diseases and pest control.

The detailed explanation of the data gathering procedure emphasises the collection of extensive datasets about crop diseases and pests. Data quality was crucial for training the developed deep learning model, and this was highlighted by discussing the preparation methods used to clean, format, and augment the data.

Justifying the model's appropriateness for decision support in crop management, this chapter explains the specific deep learning architecture used in this research. It delves into the training process, including areas like validation procedures and hyperparameter tuning, to guarantee the model's generalizability and dependability. Transparency in the assessment approach was further enhanced by outlining the evaluation measures and criteria that were used to analyse the model's performance.

In order to make sure the research was ethical, the researcher addressed the ethical concerns by including how to utilise data responsibly and whether the model has any biases. The chapter stresses that the methodology was in line with the research objectives and that the study helped farmers by developing a deep learning model that helped with crop diseases and pest management decisions.

CHAPTER FOUR
DEEP LEARNING MODEL FACTORS INFLUENCING DECISION SUPPORT IN CROP DISEASES AND PEST MANAGEMENT

4.1 Introduction

This chapter highlights the researcher's emphasis on the growing use of deep learning models to improve decision support systems, with a focus on crop diseases and pest management. These models improved agricultural productivity and reduced economic losses by accurately classifying crop diseases and pests and recommending early treatments using large datasets and deep learning algorithms.

In this chapter, the researcher looked at data quality, model design, computing resources, and other factors that affected how well these deep learning models performed. It brought to light the difficulties of gathering thorough and reliable data from varied agricultural settings. Also, picking the right model architectures to deal with the complexity and unpredictability of data was emphasised. The need to have a strong computer network to handle and analyse massive datasets was also emphasised.

The research laid the groundwork for a comprehensive analysis of these factors' effects on the effectiveness of deep learning models in actual agricultural contexts. Its purpose was to set the stage for the study's analysis and results, which shed light on agricultural technology for academics and professionals [237].

Algorithm 1: Key Factor Identification for Decision Support in Crop Diseases and Pest Management Using Deep Learning

Objective:
To identify and analyze key factors influencing decision support in crop diseases and pest management using deep learning models

Input:
- Crop disease and pest dataset, CropDataset
- Initial deep learning models, ModelList (e.g., CNN, DenseNet, AlexNet, EfficientNet, VGG)

Output:
- Key decision support factors, KeyFactors
- Optimal model and configuration, OptimalModel

Procedure:

1. Data Collection and Preprocessing:
(a) Gather image dataset of healthy crops, diseases, and pests, CropDataset
(b) Preprocess images:
- Resize and normalize pixel values
- Perform data augmentation (rotation, flipping, scaling)

(c) Split the dataset into training, validation, and testing sets

2. Model Selection and Initialization:
(a) Define deep learning model architectures in ModelList
(b) Initialize each model with parameters based on architecture (layer depth, filter size, etc.)

3. Model Training:
For each model in ModelList:
(a) Compile model with loss function (e.g., categorical cross-entropy) and optimizer (e.g., Adam)
(b) Train model on CropDataset (training set) with validation set evaluation:
- For each epoch:
  i. Forward propagate images through the model
  ii. Calculate loss and update weights via backpropagation
- Track validation accuracy and save the best-performing model for each architecture

4. Model Evaluation and Comparison:
(a) Evaluate each trained model on the test set using accuracy, precision, recall, and F1-score
(b) Identify best-performing model in terms of accuracy and generalizability, noted as OptimalModel

5. Hyperparameter and Model Parameter Analysis:
(a) Investigate effects of key model parameters (e.g., layer depth, filter size, learning rate)
(b) Record how each parameter impacts classification accuracy and robustness

6. Identify Key Decision Support Factors:
(a) Based on model parameter analysis, identify key decision support factors:
- Note specific model configurations (e.g., filter sizes)

(b) Document parameters and factors contributing most to decision support outcomes

7. Results and Recommendations:
(a) Summarize insights from parameter analysis, highlighting KeyFactors for decision support
(b) Recommend OptimalModel and best practices for model configuration and data preprocessing for crop disease management

Output:
- KeyFactors: List of critical configurations influencing decision support
- OptimalModel: Best model configuration for crop disease and pest management decision-making

Figure 4.1 Key Factor Identification for Decision Support in Crop Diseases and Pest Management Using Deep Learning

By using deep learning models to examine image data, the algorithm in Figure 4.1 sought to discover and examine critical aspects that impact decision support in controlling crop diseases and pests. First, a collection of images labelled as healthy or diseased was collected and preprocessed. To improve the generalizability of the model, the images were preprocessed by resizing, normalising, and augmenting. For the purpose of training and evaluating the model, the data was then divided into training, validation, and testing subsets.

Before conducting the experiments, a number of deep learning models were chosen and set up, such as CNN, DenseNet, AlexNet, EfficientNet, and VGG. The architecture-specific characteristics, including layer depth and filter sizes, were used to initialise each model. Models were built during training using the Adam optimiser to update weights through backpropagation and categorical cross-entropy as the loss function. Each model's best performance was recorded, and training accuracy was assessed by validating on a different data set [238].

Following training, the accuracy, precision, recall, and F1-score of each model were assessed on the test data set. These assessment criteria and the model's generalizability across different test samples were used to identify the top-performing model, which was called the Optimal Model. The researcher also examined the effect of model parameters and hyperparameters (such as learning rate, filter size, and layer depth) on classification accuracy and flexibility to examine factors that contributed to decision support effectiveness [239].

This study determined critical decision support components by analysing the effects of various setups and model parameters on model performance. The results, which included important settings like filter dimensions and layer depth, were recorded as key factors that greatly impacted the results of decision-making about pest and disease control in crops. To conclude, the researcher proposed the optimum approach to model design, data preparation, and model setup for improved decision support for crop disease and pest management. The best configuration for decision support for crop disease and pest management was represented by the optimal model, and the result was a list of key factors [240].

4.2 DenseNet

The accuracy of image classification in crop disease and pest management was greatly enhanced by the optimum propagation and usage of features using dense connections and DenseNet architectural components. Notably, DenseNet improved gradient flow and reduced the vanishing gradient problem in deep networks by constructing its layers so that each layer gets input from all the preceding layers [241]. A mathematical representation of the dense connectedness was;

$$H_1 = H_1([X_0, X_1, ..., X_{n-1}]) \quad \text{(Eq. 4.1)}$$

The output of the l-th layer was represented by x_l, which was a composite function of batch normalisation, ReLU, and Convolution. The set $[x_0, x_1, ..., x_{n-1}]$ represented the sum of the feature mappings from all the preceding layers.

The number of filters added to every layer was determined by the growth rate k, which was an important factor. The model could capture different characteristics without a high computational cost due to a 32 growth rate increment of the number of feature maps. Here is the formula for the feature maps that were produced at layer l;

$$F_l = F_0 + l \times k \quad \text{(Eq. 4.2)}$$

where the layer index was represented by l, and the number of input feature maps was denoted by F_0.

By using 1x1 convolutions as bottleneck layers, the researcher was able to reduce the number of input feature maps before applying 3x3 convolutions, which improved efficiency and reduced computational cost. Another way to describe the compression factor $\boldsymbol{\theta}$ is as follows;

$$F_{out} = \theta F_{in} \quad \text{(Eq. 4.3)}$$

$\boldsymbol{\theta}$ was set to 0.5, and F_{out} and F_{in} were the numbers of output and input feature maps, respectively. The researcher was able to keep the model size moderate while keeping crucial information due to this compression.

Applying dropout regularisation at a rate of 0.5 improved the model's generalizability by reducing the likelihood of overfitting via the random deactivation of neurons during training. The model's capacity to distinguish between healthy and crop diseases or

identification of particular pests was enhanced by the extraction of complex and hierarchical features from crop and pest images, which was made possible by the network's depth, as shown by DenseNet layers [242].

Together, these factors allowed for more effective feature propagation, reduced computing complexity, and prevented overfitting, all of which led to improved classification accuracy in crop diseases and pest management. Because of this, a solid and trustworthy decision-support system for crop diseases and pests was created [243].

Table 4.1 DenseNet Architectural Components.

Component	Description	Value Used	Influence on Decision Support
conv0	Initial convolution	3→64 channels, 7×7, stride 2	Extracted basic image features such as edges and textures.
norm0	Batch normalization	64 channels	Stabilised training and improved accuracy.
relu0	Activation function	ReLU	Enabled learning of complex disease and pest patterns.
pool0	Max pooling	3×3, stride 2	Reduced image size and computational cost.
denseblock1	First DenseBlock	6 DenseLayers	Improved feature reuse and fine-detail extraction.
denselayer1	First dense layer	1×1 and 3×3 convolutions	Learned low- and high-level image features.
denselayer2	Second dense layer	Input: 96 channels	Expanded features for more detailed pattern detection.
denselayer3	Third dense layer	Input: 128 channels	Combined earlier features to detect subtle symptoms.
denselayer4	Fourth dense layer	Input: 160 channels	Improved recognition of complex disease symptoms.
denselayer5	Fifth dense layer	Input: 192 channels	Distinguished diseases and pests with similar symptoms.
denselayer6	Sixth dense layer	Input: 224 channels	Refined features and improved classification accuracy.

A densely connected convolutional network, in which each layer received inputs from all previous layers, was the basis of the DenseNet design used for image classification in crop diseases and pest management. With this layout, it was easier to reuse features, which increased efficiency and precision. The model was able to extract, normalise, and activate crucial low-level image features with the help of components such as the original Conv2d, BatchNorm, and ReLU. By iteratively learning intricate patterns of

crop diseases and pests, DenseBlocks, which were composed of several DenseLayers, propagated these features forward. Classification accuracy was much improved as a result of the network's ability to create hierarchical feature representations, made possible by the expansion of feature channels across layers. Faster training convergence and increased accuracy were brought about by the densely connected layers, which helped to reduce the vanishing gradient problem. This was particularly true in images where there were minor distinctions between healthy, affected crops and pests [244].

Using fine-tuned components, the DenseNet architecture significantly improved the accuracy of image classification for crop diseases and pest control. From the input images, the first convolution layer (conv0) extracted basic features, including edges and textures. These features laid the groundwork for more sophisticated crop diseases and pest classification. The configuration of this layer was Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False). Batch normalisation was then used, using the parameters BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True), to standardise the inputs to the subsequent layers. This stabilised the training process by preventing the gradients from vanishing. Performing this normalisation step enhanced the model's training speed and accuracy. The network was able to learn intricate patterns in the dataset, such as the small visual distinctions between healthy, affected crop leaves and pests, due to the non-linearity introduced by the ReLU(inplace=True) activation function. The feature maps were also reduced in spatial dimensions by the max pooling layer (pool0) with kernel_size=3, stride=2, padding=1. This made the model more computationally efficient while keeping the most relevant information.

The architecture's dense blocks, which were made up of dense layers, were able to capture more abstract features when they combined lower-level features with newly extracted features. With the use of two convolution operations, one with kernel_size=(1, 1) and another with kernel_size=(3, 3), applied to each layer of the dense block, the model was able to improve feature extraction at all scales. An increase in the network's capacity to generalise to different crop diseases and pests was brought about by the fact that feature reuse was made easier when feature dimensions were increased at each layer. The model's ability to accurately capture and distinguish

between complex visual elements in images of crop diseases and pests was enhanced by these design selections, which in turn led to its high classification accuracy [245].

4.3 AlexNet

AlexNet's use of a deep convolutional network to efficiently extract hierarchical features from crop leaf diseases and pests images improved the accuracy of image classification in the context of pest control and crop diseases. In order to increase gradient flow and create non-linearity, the design used five convolutional layers followed by three fully connected layers. The convolutional layers used Rectified Linear Unit (ReLU) activation functions. This was the definition of the ReLU activation function;

$$f(x) = \max(0, x) \quad \text{(Eq. 4.4)}$$

The network was able to acquire intricate patterns necessary for differentiating between healthy and affected crops because of its non-linearity, which helped it overcome the vanishing gradient problem.

To pick up low-level features like textures and edges, the first two convolutional layers used 11x11 and 5x5 receptive fields, respectively. The following was the formula for AlexNet convolution operation;

$$Y_{i,j,k} = \sum_{m,n} x_i + m, j+n, k.w_{m,n,k} + b_k \quad \text{(Eq. 4.5)}$$

The variables were defined as: $Y_{i,j,k}$ represented the output feature map, x for the input image, w for the convolution kernel, b_k for the bias, and m and n for the kernel dimensions. The network was able to record intricate spatial hierarchies due to these wide receptive fields.

The feature maps were reduced in spatial dimensions by max-pooling layers that followed the first, second, and fifth convolutional layers. This reduced the computational cost and prevented overfitting. The max pooling procedure was expressed as;

$$Y_{i,j,k} = \max_{m,n}(x_i + m, j+n, k) \quad \text{(Eq. 4.6)}$$

Where $max_{m, n}$ represented the highest value that was found inside the pooling set. The number of parameters was reduced, yet the most significant features were retained by the max pooling operation.

By randomly deactivating neurons during training, AlexNet used dropout regularisation with a 0.5 dropout rate in the fully connected layers to prevent overfitting. The model's ability to generalise was enhanced by this strategy, which made it more resilient when subjected to unseen data.

By varying the training data and using data augmentation techniques like random cropping and flipping of the input images, the researcher was able to improve generalisation and further boost the model's flexibility and accuracy.

The fully connected layer performed the final classification by combining the learned features into a strong representation, while the softmax function transformed the logits into probabilities. Here is the definition of the softmax function;

$$P(y = j \mid x) = \frac{e^{z_j}}{\sum_k e^{z_k}} \qquad \text{(Eq. 4.7)}$$

z_j represented the logits for every class j. The identification of crop diseases and pests was made easier with the help of this probabilistic output.

The architectural components of AlexNet, including large receptive fields in convolutional layers, ReLU activation functions, max-pooling layers, dropout regularisation, and data augmentation techniques, collectively enhanced feature extraction, reduced overfitting, and improved generalisation, leading to improved image classification accuracy in crop diseases and pest management. As a result, diagnosis became more accurate and trustworthy, which in turn led to enhanced farming practices [246].

Table 4.2 AlexNet Architectural Components.

Component	Configuration	Influence on Decision Support
Conv2d (0)	3→64 channels, 11×11 kernel, stride 4	Extracted basic features such as edges and colours for disease recognition.
ReLU (1)	ReLU activation	Enabled learning of complex image patterns.
MaxPool2d (2)	3×3 kernel, stride 2	Reduced feature dimensions while preserving key information.
Conv2d (3)	64→192 channels, 5×5 kernel	Extracted higher-level features for disease differentiation.
ReLU (4)	ReLU activation	Improved learning of non-linear relationships.
MaxPool2d (5)	3×3 kernel, stride 2	Reduced feature size and improved efficiency.
Conv2d (6)	192→384 channels, 3×3 kernel	Captured deeper disease and pest patterns.
ReLU (7)	ReLU activation	Supported learning of complex pest features.
Conv2d (8)	384→256 channels, 3×3 kernel	Enhanced feature extraction for accurate classification.
ReLU (9)	ReLU activation	Improved recognition of disease symptoms.
Conv2d (10)	256→256 channels, 3×3 kernel	Learned advanced features for similar disease classes.
ReLU (11)	ReLU activation	Strengthened pattern learning.
MaxPool2d (12)	3×3 kernel, stride 2	Reduced feature maps before classification.
AdaptiveAvgPool2d	Output size 6×6	Standardised feature dimensions for the classifier.
Dropout (0)	p=0.5	Reduced overfitting and improved generalisation.
Linear (1)	9216→4096 features	Learned complex feature relationships.
ReLU (2)	ReLU activation	Added non-linearity to classification layers.
Dropout (3)	p=0.5	Improved model robustness.
Linear (4)	4096→4096 features	Increased classification capability.
ReLU (5)	ReLU activation	Enhanced feature representation.
Linear (6)	4096→1000 classes	Produced final classification outputs adapted to crop disease and pest categories.

The structured components of the AlexNet architecture, which were created to efficiently extract and learn features from images, had a substantial impact on decision support in crop diseases and pest management.

The crucial part of feature extraction was performed by the first layer, Conv2d(3, 64, kernel_size=(11, 11), stride=(4, 4), padding=(2, 2)), which processed the input images (which had three colour channels) and generated 64 feature maps. In the early phases of image processing, the 11x11 kernel size was big enough to extract larger elements like textures and edges. The layer improved the performance of the next layers by reducing the spatial dimensions of the input image with a stride of 4. This allowed for faster calculations while preserving vital information. The output size was kept under control by a padding of 2, which prevented a significant decrease in spatial resolution. The framework for finding critical features associated with crop diseases and pests was laid out during this first phase.

The model's non-linearity was introduced using the ReLU(inplace=True) activation function, which followed the convolutional layer. With the help of ReLU, AlexNet was able to learn intricate patterns that indicated different types of crop diseases and pests, all because the network could take into consideration non-linear features. Due to the presence of nuanced differences in crop diseases and pests datasets, this phase was critical for ensuring correct classification.

With the feature maps down-sampled by the MaxPool2d(kernel_size=3, stride=2, padding=0) layer, the computational cost was reduced, but the most relevant features were retained. This pooling procedure improved feature abstraction, which in turn made the model more resistant to crop diseases and pest image variations by increasing its invariance to small translations in the input data.

A second convolutional layer, Conv2d(64, 192, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2)), took the output from the first layer and expanded it into 192 feature maps, enhancing the model's power to identify more complex features. The model was able to shift its attention to more general features, such as unusual development patterns or discolouration, which were crucial for identifying crop diseases and pests.

Successive ReLU activations, such as ReLU(inplace=True), enhanced the extraction of non-linear features even more. To further abstract the features and aid in

dimensionality reduction, an additional MaxPool2d(kernel_size=3, stride=2, padding=0) layer was added.

Additional convolutional layers were included in the design, including Conv2d(192, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1)) and Conv2d(384, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1)), which were subsequently activated by ReLU. In order for the model to be able to differentiate between different types of crop diseases and pests, these layers gradually extracted more complex features. Due to the complex patterns found in the images, which needed complex representation learning, the model learnt complex features for the classification to be correct.

The data was prepared for the next fully connected layers by the last pooling layer, MaxPool2d(kernel_size=3, stride=2, padding=0), which further reduced the spatial dimensions and extracted the most significant features for the classification task.

Regardless of the input dimensions, the AdaptiveAvgPool2d(output_size=(6, 6)) layer successfully standardised the output size of the feature maps. A smooth handoff to the classifier was made possible by this standardisation, which was crucial for keeping the model's output consistent.

To avoid overfitting, the classifier layer started with a Dropout(p=0.5) layer that randomly disabled 50% of the neurons during training. This method improved the model's generalizability when applied to real-world scenarios by preventing it from being unduly dependent on particular neurons.

The first fully connected layer, Linear(in_features=9216, out_features=4096), increased the dimensionality of the pooled feature maps, which helped the model understand the intricate feature relationships necessary for crop diseases and pest identification. To improve the model's capacity to learn intricate patterns, a second ReLU(inplace=True) layer was added, which introduced another degree of nonlinearity.

To further reduce the possibility of overfitting, another Dropout(p=0.5) layer was added. After this came Linear(in_features=4096, out_features=4096), a second fully connected layer that greatly enhanced the feature representation. The last linear layer of the design, Linear(in_features=4096, out_features=1000), produced results that corresponded to the classes in the classification task. Improving decision support in

agricultural tasks was possible through fine-tuning AlexNet for a particular number of crop diseases and pest classes, even if it was initially developed for 1000 classes.

The model was able to identify complex patterns linked to various crop diseases and pests because of its layered architecture, which allowed for excellent feature extraction and representation learning. Improved accuracy in recognising and classifying crop diseases and pests was achieved by the careful use of activation functions, pooling layers, and dropout techniques. This, in turn, helped farmers and agricultural stakeholders make better decisions [247].

4.4 EfficientNet

Through the use of a well-balanced model scaling strategy that optimised depth, width, and resolution, EfficientNet architectural components had a substantial impact on the decision support in crop disease and pest management. To improve speed without dramatically increasing computational complexity, EfficientNet used a compound scaling algorithm that evenly scaled these three dimensions. Here are the formulae that described the scaling method;

Depth: $d=\alpha^i$

Width: $w=\beta^i$

Resolution: $r=\gamma^i$

Where the scaling factor was denoted by I, and the constants α,β, and γ were found by a grid search. The model was able to learn complex features from crop disease and pest images because EfficientNet was able to maintain a balanced increase in network capacity.

When compared to regular convolutions, EfficientNet's use of depthwise separable convolutions reduced computational cost and parameter count. As shown technically, a depthwise separable convolution was composed of depthwise and pointwise convolutions.

$$\text{Depthwise convolution; } y_{i,j,k} = \sum_{m,n} x_i + m, j+n, k.w_{m,n,k} \quad \text{(Eq. 4.8)}$$

$$\text{Pointwise convolution; } y'_{i,j,q=\sum_k y_{i,j,k}}.w'_{k,q} \quad \text{(Eq. 4.9)}$$

where the outputs of the depthwise and pointwise convolutions were represented by y and y' respectively, the input feature map was denoted by x, and the convolution kernels were represented by w and w'.

Furthermore, EfficientNet included Squeeze and Excitation (SE) blocks that dynamically realigned feature outputs on a channel-by-channel basis. The model's sensitivity to useful features was improved by SE blocks, while less helpful features were suppressed. The operation of the SE block was defined as;

$$s = \sigma(W_2.\mathrm{ReLU}(W_1.z)).x \qquad \text{(Eq. 4.10)}$$

σ was the sigmoid function, x was the input feature map, z was the global average pooled feature map, and W_1 and W_2 were the weights of the fully connected layers in the SE block. After undergoing this calibration, the model was now able to zero in on the most important features for determining whether crops are healthy or affected, as well as for identifying pests.

Another factor that enhanced EfficientNet's accuracy was its use of the swish activation function. Here is the definition of the swish function used;

$$f(x) = x.\sigma(x) \qquad \text{(Eq. 4.11)}$$

where the sigmoid function was represented by σ. The smooth non-linearities produced by this activation function improved training convergence by increasing gradient flow.

Improved crop diseases and pest management classification accuracy was achieved by EfficientNet with the combined efforts of compound scaling, SE blocks, depthwise separable convolutions, and swish activation. Supporting better decision-making and agricultural practices, these components enabled the model to learn and use complex features effectively, leading to more accurate and reliable diagnosis of crop diseases and pests [248].

Table 4.3 EfficientNet Architectural Components.

Component	Value	Influence on Decision Support
Conv2d	3→48 channels, 3×3 kernel, stride 2	Extracted basic features such as edges and textures while reducing computation.
BatchNorm2d	48 channels	Stabilised training and improved model generalisation under different conditions.
SiLU Activation	SiLU	Added smooth non-linearity, improving learning of subtle disease and pest patterns.
MBConv Block	Depth-wise Conv2d (48→48)	Efficiently captured detailed leaf features such as spots and lesions.
Squeeze-Excitation	Conv2d(48→12→48)	Highlighted important disease regions and improved feature selection.
AdaptiveAvgPool2d	Output size = 1	Compressed features and improved robustness across disease categories.
StochasticDepth	p=0.0051	Reduced overfitting and improved performance on unseen images.
Conv2d (Second Block)	Depth-wise Conv2d (24→24), 3×3 kernel	Captured fine variations in colour and texture for disease differentiation.
Squeeze-Excitation (2nd)	Conv2d(24→6→24)	Further refined important features for accurate classification.
BatchNorm2d (2nd)	24 channels	Maintained stable learning and improved adaptation to image variations.

The EfficientNet design enhanced decision support for crop diseases and pests by combining many different approaches. Using the above parameters, the model's first convolution layer was able to effectively extract low-level features like edges and textures from the input crop diseases and pest images. Reducing computing complexity without forfeiting crucial visual information related to crop diseases and pests was achieved by downsampling the input image with a stride value of 2. To keep training stable and avoid problems like vanishing gradients, the researcher used BatchNorm2d(48) to normalise the feature maps. Because of this, the model was able to generalise better across different crop leaf disease and pest images, even when subjected to different levels of noise and brightness.

A smooth non-linearity was added to the model by using the Sigmoid-Weighted Linear Unit-SiLU(inplace=True) activation function in the first layer. This allowed the model to learn small but important differences in crop leaf disease symptoms and pests. Increased accuracy in crop diseases and pests was achieved by using SiLU's smoothness to store and analyse even small changes in the input image, such as little spots or discolourations on leaves.

The model used a series of Mobile Inverted Bottleneck Convolution (MBConv) blocks that started with Conv2d(48, 48, kernel_size=(3, 3), groups=48, bias=False) and ended with depth-wise separable convolutions. The model's computational performance was greatly enhanced by these layers, which divided the convolution process across several feature maps. This allowed for a rapid and lightweight model, which was especially useful for extracting complex disease patterns from leaves, such as lesions and rusts. Various feature maps were re-emphasised by the SqueezeExcitation (SE) module that came after these blocks. Using AdaptiveAvgPool2d(output_size=1) and two fully connected layers (Conv2d(48, 12) and Conv2d(12, 48)), for instance, it adaptively highlighted disease-affected areas in the image, allowing the model to focus on the most important symptoms for pest and crop disease identification.

A random layer-dropping technique called stochastic depth with mode=row was also part of the design (p=0.005128205128205128, mode=row). By making the model learn more generic features, this regularisation strategy increased its flexibility and enhanced its performance on crop disease and pest images that were previously undetected. The model's consistent usage of batch normalisation layers also contributed to stable gradient flow, which prevented the network from overfitting to the training data and allowed it to function well in a practical agricultural environment.

The EfficientNet's ability to effectively and efficiently classify crop diseases and pests was a result of its architectural components, which included features like SqueezeExcitation blocks, smooth activation functions like SiLU, and depth-wise convolutions. The model's capacity to extract important features, such as areas of leaves affected by a disease, led to enhanced decision support for agricultural stakeholders and enabled more accurate and faster identification of crop diseases and pests. In the end, this helped farmers in all sorts of different regions to make better decisions about crop diseases and pest control [249].

4.5 Visual Geometry Group

By using a deep convolutional network with a standard architecture that allowed for precise feature extraction, the Visual Geometry Group (VGG) architectural components greatly affected the accuracy of image classification in crop diseases and pest management. In order for the network to acquire the fine-grained features necessary for differentiating between healthy and affected crops or detecting particular pests, the VGG model made use of stacked convolutional layers with small receptive fields (3x3). The following was the formula for a VGG convolution operation;

$$Y_{i,j,k} = \sum_m \sum_n x_i + m, j + n, k.w_{m,n,k} \quad \text{(Eq. 4.12)}$$

where $Y_{i,j,k}$ represented the feature map that was produced, x for the image that was the input, w for the convolution kernel, and m and n for the dimensions that were used by the kernel. By extending the receptive field and enabling hierarchical feature learning, the VGG network's depth, achieved by stacking 16 to 19 weight layers, improved the model's capacity to grasp complex patterns. To accurately classify crop diseases and pests, it was necessary to extract both low-level features (like edges and textures) and high-level features (like forms and object segments), and this depth made it possible. In order to reduce the spatial dimensions of feature maps while keeping the most relevant features, VGG also used max-pooling layers, which were represented as;

$$Y_{i,j,k} = \max_{m,n}(x_i + m, j + n, k) \quad \text{(Eq. 4.13)}$$

Where the greatest value inside the pooling window was denoted by $\max_{m,n}$. This method of pooling introduced translation invariance, which helped reduce computational complexity and avoid overfitting. The VGG used fully connected layers at the network's edge to classify the features that had been extracted. The combination of the learned features into a strong representation of the classification problem was greatly achieved by these layers.

One reason VGG was so easy to understand and apply was because of its consistent architecture, which was defined by the recurrent application of small convolutional filters. Because of this design choice, VGG was a solid option for image classification tasks, including crop leaf diseases and pests, and it was easy to apply and fine-tune.

In general, the architectural components of VGG, including fully connected layers, max pooling, and deep convolutional layers with small receptive fields, enabled high feature extraction and robust classification, which in turn supported better decision-making and agricultural practices in the context of crop diseases and pest management [249], [250].

Table 4.4 VGG Architectural Components.

VGG Component	Exact Value	Influence on Decision Support
Conv2d (Early Layers)	3→64 and 64→64 channels, 3×3 kernel	Extracted basic features such as edges, textures, and leaf patterns.
ReLU Layers	Inplace=True	Added non-linearity for learning complex disease and pest patterns.
MaxPool2d (Early)	2×2 kernel, stride 2	Reduced feature size while preserving important disease information.
Conv2d (Middle Layers)	64→128 and 128→128 channels, 3×3 kernel	Captured intermediate features such as spots, discoloration, and symptom patterns.
Conv2d (Advanced Layers)	128→256 and 256→256 channels, 3×3 kernel	Learned deeper disease characteristics and severity indicators.
Conv2d (Deep Layers)	256→512 and 512→512 channels, 3×3 kernel	Extracted highly detailed features for distinguishing complex diseases and pests.
MaxPool2d (Deep Layers)	2×2 kernel, stride 2	Reduced feature complexity while retaining critical disease markers.
AdaptiveAvgPool2d	Output size (7×7)	Standardised feature maps for classification and improved generalisation.
Linear Layer 0	25088→4096 features	Converted extracted image features into classification representations.
ReLU (Classifier)	Inplace=True	Improved learning of complex relationships between features and diseases.
Dropout Layers	p=0.5	Reduced overfitting and improved performance on unseen crop images.
Linear Layer 3	4096→4096 features	Refined feature relationships for accurate disease classification.
Linear Layer 6	4096→1000 classes	Produced final predictions for crop disease and pest categories.

The important features of the VGG architecture that were used for crop diseases and pest control affected how well it worked in the decision support system. Using a 3x3

kernel size, a stride of 1, and a padding of 1, the architecture started with Conv2d layers that processed the input images. The first layer took three input channels, which represented RGB images, and produced 64 feature maps. Modelling fine-grained features like colour variations, leaf textures, and disease lesions was made possible by these arrangements. The Rectified Linear Unit (ReLU) activation added non-linearity to the model, enabling it to learn more intricate patterns from the image data.

The Conv2d layers, ReLU activations, and MaxPool2d layers made up each convolutional block in VGG. For instance, after the two initial 64-channel Conv2d layers, the feature maps' spatial dimensions were reduced in half using a MaxPool2d layer with a 2 kernel size and a 2 stride. The computational complexity was reduced, and the model was able to extract distinct features from crop disease spots and pest images. Eventually, the network was able to extract more abstract and high-level features from the images with a total of 512 filters. To accurately identify a wide range of crop diseases and pest infestations, deep feature extraction was important.

Deeper in the network, the 512-channel Conv2d layers were able to capture subtle features like the colour, texture, and shape of diseased crop regions. Subtle alterations in crop leaf structure and discolouration were important signs in detecting more advanced disease stages or complicated pest patterns, which is why this degree of feature extraction was significant. By adding MaxPool2d layers after each pair of Conv2d and ReLU layers, the spatial dimensions were further reduced. This allowed the retention of the most relevant disease-related features while minimising the noise.

The feature extraction process concluded with an AdaptiveAvgPool2d layer that averaged the feature maps' spatial dimensions; its output size was (7,7). By combining all of the learned features into a single set, the model was able to generalise and retain the most relevant features on crop diseases and pests for classification [251].

The AdaptiveAvgPool2d layer flattened the output to 4,096-dimensional feature vectors from 25,088 input features. The Linear layers in the classifier unit then transferred this data to the classifier for classification. When it came to understanding the connections between crop disease and pest classifications and the extracted features, these layers were vital. As a measure against overfitting, the researcher used two fully connected layers, one with 4,096 output features followed by ReLU activations and Dropout layers (with p=0.5). Particularly useful for dealing with a

wide range of affected images captured in varying conditions, the dropout layer improved the model's generalizability by randomly deactivating neurons during training.

Last but not least, the model was able to classify images into various crop diseases and pest classes by mapping 4,096 features to 1,000 output classes in the Linear layer. To facilitate early diagnosis, accurate treatment recommendations, and efficient crop disease and pest control strategies, this classification was important for giving out accurate decision support to stakeholders.

The model's capacity to classify crop diseases and pests was greatly enhanced by the model's ability to extract full information from crop disease and pest images, which was made possible by the VGG architecture's use of pooling, dropout, fully connected layers, and deep convolutional layers. Better crop management and increased yields were the results of this feature extraction and classification process's contribution to timely and accurate decision support for stakeholders [252].

4.6 Convolution Neural Network

Efficiently collecting and analysing hierarchical features from crop disease and pest images, the architectural components of Convolutional Neural Network (CNN) greatly impacted the accuracy of image classification in crop disease and pest management. The convolutional layer, which was the main component, used kernels, which were filters, to apply convolution operations to input images to identify different features. To do a convolution, the formula was;

$$Y_{i,j,k} = \sum_m \sum_n x_i + m, j + n, k.w_{m,n,k} \quad \text{(Eq. 4.14)}$$

The output feature map was denoted as $y_{i,j,k}$, the input image was represented by x, the convolution kernel was denoted as w, and the dimensions of the kernel were denoted as m and n. This process enabled the model to recognise pests and differentiate between healthy and affected crops by capturing important patterns, including forms, textures, and edges.

To reduce computational complexity and avoid overfitting, the max pooling layer reduced the spatial dimensions of the feature maps while maintaining the most relevant features. It was possible to express the max pooling operation as;

$$Y_{i,j,k} = \max_{m,n}(x_i + m, j + n, k) \quad \text{(Eq. 4.15)}$$

where $max_{m,n}$ represented the highest value found inside the pooling set. The model became more efficient and less susceptible to overfitting as a result of this dimensionality reduction, which helped to retain the relevant features.

The model was able to learn complex patterns because of the non-linearity introduced by the Rectified Linear Unit (ReLU) activation function. The ReLU function was defined as;

$$f(x) = \max(0, x) \quad \text{(Eq. 4.16)}$$

By reducing the impact of the vanishing gradient problem, this non-linearity made it possible to train deep networks more efficiently.

The network's fully connected layers received the features that had been extracted, and this helped in the classification task. A strong representation for accurate classification was produced by combining the learned hierarchical features with the convolutional layer. The output layer, known as the softmax layer, transformed the logits into probabilities, which were represented as;

$$P(y = j \mid x) = \frac{e^{z_j}}{\sum_k e^{z_k}} \quad \text{(Eq. 4.17)}$$

z_j represented the logits for every class j. The classification of crop diseases and pests was made easier with the help of this probabilistic output.

In order to avoid overfitting, dropout regularisation was used during training to randomly deactivate a proportion of neurons. The model's ability to generalise was enhanced by this strategy, which made it more robust when exposed to unseen data.

Improving image classification accuracy in crop diseases and pest management was achieved by the synergistic action of CNN architectural components, including convolutional layers, pooling layers, ReLU activation functions, a fully connected layer, and dropout regularisation. The model was able to avoid overfitting through dimensionality reduction, introducing non-linearity for complex pattern recognition, and extracting hierarchical and detailed features, all of which contributed to more accurate and reliable diagnosis for efficient farming practices [253].

Table 4.5 CNN Architectural Components.

Component	Details	Influence on Decision Support
Conv2d Layers (1–2)	3→32 channels, 3×3 kernel	Extracted basic leaf features such as colour changes, textures, and disease spots.
ReLU + BatchNorm	Non-linear activation, ε=1e-5, momentum=0.1	Improved feature learning, training stability, and generalisation.
MaxPool2d (Early)	2×2 kernel, stride 2	Reduced feature size while preserving important disease indicators.
Conv2d Layers (3–4)	32→64 channels, 3×3 kernel	Captured more detailed pest patterns and lesion structures.
ReLU + BatchNorm (64 channels)	Non-linear activation, normalisation	Enhanced recognition of complex disease and pest features.
MaxPool2d (Middle)	2×2 kernel, stride 2	Reduced computational complexity while retaining key features.
Conv2d Layers (5–6)	64→128 channels, 3×3 kernel	Learned deeper representations of disease symptoms and leaf textures.
ReLU + BatchNorm (128 channels)	Activation and normalisation	Improved detection of subtle disease patterns and model consistency.
MaxPool2d (Deep)	2×2 kernel, stride 2	Compressed feature maps while preserving important information.
Conv2d Layers (7–8)	128→256 channels, 3×3 kernel	Extracted advanced features related to disease severity and progression.
ReLU + BatchNorm (256 channels)	Activation and normalisation	Improved classification accuracy and model robustness.
Final MaxPool2d	2×2 kernel, stride 2	Prepared condensed feature representations for classification.
Dropout	Probability 0.5 and 0.4	Reduced overfitting and improved performance on unseen images.
Linear Layer 1	50,176→1,024 features	Converted extracted image features into classification representations.
ReLU (Dense Layer)	Non-linear activation	Improved learning of complex relationships between disease features.
Linear Output Layer	1,024→4 classes	Generated final predictions for crop disease and pest categories.

Sequential convolutional and dense layers collected, processed, and classified features from images in the CNN architecture used for decision support in crop diseases and pest management. The convolutional layers were built using Conv2d. The first layer

had 32 output channels, a 3x3 kernel size, a 1x1 stride, and 1x1 padding. The three input channels represented the Red, Green, and Blue (RGB) colour channels. Colour and texture variations, which were important signs of early crop diseases, were able to be captured by the network in this setup. After every convolutional layer, the ReLU was used as its activation function to introduce non-linearity. This was necessary for the network to learn complex features that went beyond simple linear correlations.

The output was normalised with an epsilon of 1e-05 and a momentum of 0.1 through BatchNorm2d layers, which the researcher applied after each convolutional process. The stability of the model and its pace of convergence during training were both enhanced by the normalisation's reduction of internal covariate changes. The initial features were further refined by the second convolutional layer, which repeated the first configuration with 32 channels and a 3x3 kernel. To improve computational efficiency without losing pest and crop disease-specific feature information, the feature map was downsampled using the MaxPool2d layer with a 2x2 kernel and a stride of 2. This reduced spatial dimensions while retaining distinct features [254].

The third convolutional block included 64 channels, and subsequent layers raised the channel depth even more. To improve the network's ability to distinguish between related pest and crop diseases, this extension was critical for gathering mid-level features, including lesion shapes and patterns of pests and crop diseases. After this layer, another MaxPool2d operation was performed to extract important mid-level features within the feature maps. The model captured high-level abstractions, such as crop disease progression and pests, in the deeper layers by increasing the channel depth to 128 and 256 channels, respectively. This enlargement was critical for understanding the subtle differences in advanced disorders, which were not as obvious in the first layers.

By allowing classification using the extracted features, the dense layers were important in decision support. By randomly deactivating neurons during training, the initial dense layer reduced the possibility of overfitting. This was achieved by using a dropout layer with a probability of 0.5. Next came a fully connected Linear layer that took 50,176 features as input and produced 1,024 features as output. Efficient and effective pest and crop disease type classification relied on this setup, which mapped

high-dimensional convolutional features into a flattened vector. The model was able to learn intricate feature-to-feature correlations using the ReLU activation function, which improved pest and crop disease classification. To make sure the model worked well with different pests and crop disease images and settings, a second dropout layer was added with a probability of 0.4. This reduced the likelihood of overfitting even more. An output vector with four features, representing the different classes of crop diseases, was generated by the last Linear layer.

Using a gradual enhancement of image features across layers from simple colour and texture in the early levels to complex pests and crop disease patterns in the final layers, the CNN architecture successfully aided decision-making in crop disease and pest management. This improved early detection and intervention by enabling accurate and actionable crop disease and pest classification.

By seeing the network's operation as a transition from input image space to output classification space, the researcher theoretically explained how this Convolution Neural Network design affected decision support in crop disease and pest management. The equation below represents the CNN's transformation, which includes feature extraction and classification. The output was achieved by each layer progressively executing its operations [255].

Given an input image $X \in \mathbb{R}^{H\times W\times C}$ with height H, widthW, and C color channels (in this case, C=3), the CNN was represented by the function:

$$\hat{y} = f(X;\theta) = \sigma(W_2 \cdot ReLU(W_1 \cdot Flatten(MaxPool(BatchNorm(ReLU(Conv(X;\theta_{conv})))))) + b_1) + b_2) \quad \text{(Eq. 4. 18)}$$

Where;

- X: Input image with channels, height, and width.
- θ: Collection of parameters (weights and biases) in the CNN.
- $Conv(X;\theta_{conv})$: Convolution layers applied to X with filters learned for feature extraction.
- ReLU: Non-linear activation function.
- BatchNorm: Batch normalization, stabilized inputs to each layer.
- MaxPool: Max pooling, reduced spatial dimensions.
- Flatten: Converted 2D feature maps into a 1D vector for the dense layers.
- W_1 and b_1 : Weights and biases in the first dense (fully connected) layer.
- W_2 and b_2 : Weights and biases in the final dense layer, mapping to output classes.
- σ: Softmax function, which converted outputs into probabilities for each class.

Explanation of Terms and Layers in the Equation

A set of convolution techniques was used to extract spatial features using the Conv(X; θ_{conv}) term in the Convolution Layers (Conv). Here, θ_{conv} contained all the biases and kernel weights across the convolution layers, which ranged from 3 to 256 channels. Then, every convolutional layer was represented by the formula;

$$h_{i,j} = \sum_{m=1}^{k}\sum_{n=1}^{k} x_{i} + m, j + n.w_{m,\, n\, +\, b} \qquad \text{(Eq. 4.19)}$$

using a kernel size of k=3, a padding of p=1, and a stride of s=1, which retained spatial dimensions while extracting hierarchical features.

The ReLU activation was performed after every convolution, using ReLU(x)=max(0,x). Learned complex feature mappings were crucial for differentiating pest and crop leaf disease features, and the ReLU activation function introduced nonlinearity.

After every convolution layer, batch normalisation was done, which was defined as;

$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \in}} \qquad \text{(Eq. 4.20)}$$

By reimbursing for variations in image brightness, contrast, and other real-world noise, this normalised the outputs to speed up convergence and improve stability.

The pooling layer reduced the spatial dimensions while retaining significant features, a process known as Max Pooling. To abstract high-level patterns like the morphologies of crop disease-affected regions or pest contours, MaxPool with a 2×2 kernel size and a stride of 2 reduced the feature map size by half after each pooling layer.

The 2D feature maps were flattened into a 1D vector by using the flatten and dense layers operations. To avoid overfitting, the dense layer calculated a linear transformation and applied a dropout.

$$\text{Flatten} \rightarrow W_1 \cdot x + b_1 \rightarrow \text{ReLU} \rightarrow W_2 \cdot x + b_2 \rightarrow \sigma \qquad \text{(Eq. 4.21)}$$

The softmax function, denoted by σ, produced a probability for each of the classes for the crop leaf diseases and pests.

The CNN was able to correctly identify and distinguish crop diseases and pest features because of its design, which included layers that gradually improved significant patterns through feature extraction, normalisation, and pooling. The class

probabilities, aided in crop disease and pest identification, and the predictions produced by the softmax output enabled informed decision-making. The model was further improved to operate effectively across varied settings using regularisation layers (BatchNorm and Dropout), which improved its practical reliability in real-world crop disease and pest management applications [256].

4.7 Summary

The decision support in crop diseases and pest management was greatly affected by Convolution Neural Network architectural components, including DenseNet, AlexNet, EfficientNet, VGG, and standard CNN. The model's capacity to learn intricate features from crop disease and pest images was improved by DenseNet's dense connection, which allowed for wide feature propagation, reducing feature loss. Dropout regularisation improved classification accuracy by reducing overfitting; AlexNet's max pooling, ReLU activation, and deep convolutional layers efficiently extracted hierarchical features and sped up training. To extract detailed features with fewer parameters, EfficientNet's compound scaling balanced the model's depth, width, and resolution. Additionally, depthwise separable convolutions and squeeze-and-excitation blocks enhanced efficiency and focused on key features. Easy to develop and tune, VGG's design made use of small convolutional filters, max-pooling layers, and ReLU activation to provide robust classification and fine-grained feature learning. Convolutional layers, max pooling, activation functions, fully connected layers, and dropout regularisation were all Convolution Neural Network components that, when combined, improved feature extraction, reduced dimensionality, added non-linearity for complex pattern recognition, and prevented overfitting. Better decision-making and agricultural practices were supported by the high classification accuracy that resulted from these enhancements across diverse architectures, which provided reliable tools for classifying crop diseases and pests.

CHAPTER FIVE
A HYBRID DEEP LEARNING MODEL FOR DECISION SUPPORT IN CROP DISEASES AND PEST MANAGEMENT

5.1 Introduction

This chapter covers the development of an enhanced deep neural network model, with an emphasis on how it uses Convolutional Neural Network (CNN) and Support Vector Machine (SVM) to correctly identify and classify pests and crop diseases. Using the best features of Convolutional Neural Network (CNN) and Support Vector Machine (SVM), this novel technique sought to build a powerful decision support system for crop diseases and pest management. Automatic feature extraction and learning using Convolutional Neural Network (CNN) was used for pests and crop disease images, enabling the identification of complex patterns and spatial hierarchies linked to various pests and crop diseases. The Support Vector Machine was used as a classifier to accurately distinguish between distinct crop diseases and pest classes using the features extracted by the Convolution Neural Network. The Support Vector Machine was well-known for its strong classification performance and capacity to manage intricate decision boundaries. Aiming to enhance classification accuracy and robustness, this hybrid model combined CNN's feature extraction abilities with the classification power of the Support Vector Machine. This chapter examines the training and validation process of the hybrid model using a large dataset of crop diseases and pest images. It shows how the model was used for accurate identification and quick interventions in crop diseases and pest management. Incorporating this enhanced model showed great promise in improving decision support systems, which led to better and more long-lasting approaches to crop diseases and pest management [257].

5.2 Deep Learning Model

In order to properly detect and classify crop diseases and pests, the improved deep neural network model integrated a Convolutional Neural Network (CNN) with Support Vector Machines (SVM). The goal of this approach was to improve the accuracy and robustness of the decision support system for crop diseases and pest management by combining the best features of Convolutional Neural Networks (CNNs) and Support Vector Machines (SVM). To make use of CNN's image feature

extraction abilities and SVM's strong classification capabilities, CNN and SVM were developed to work together.

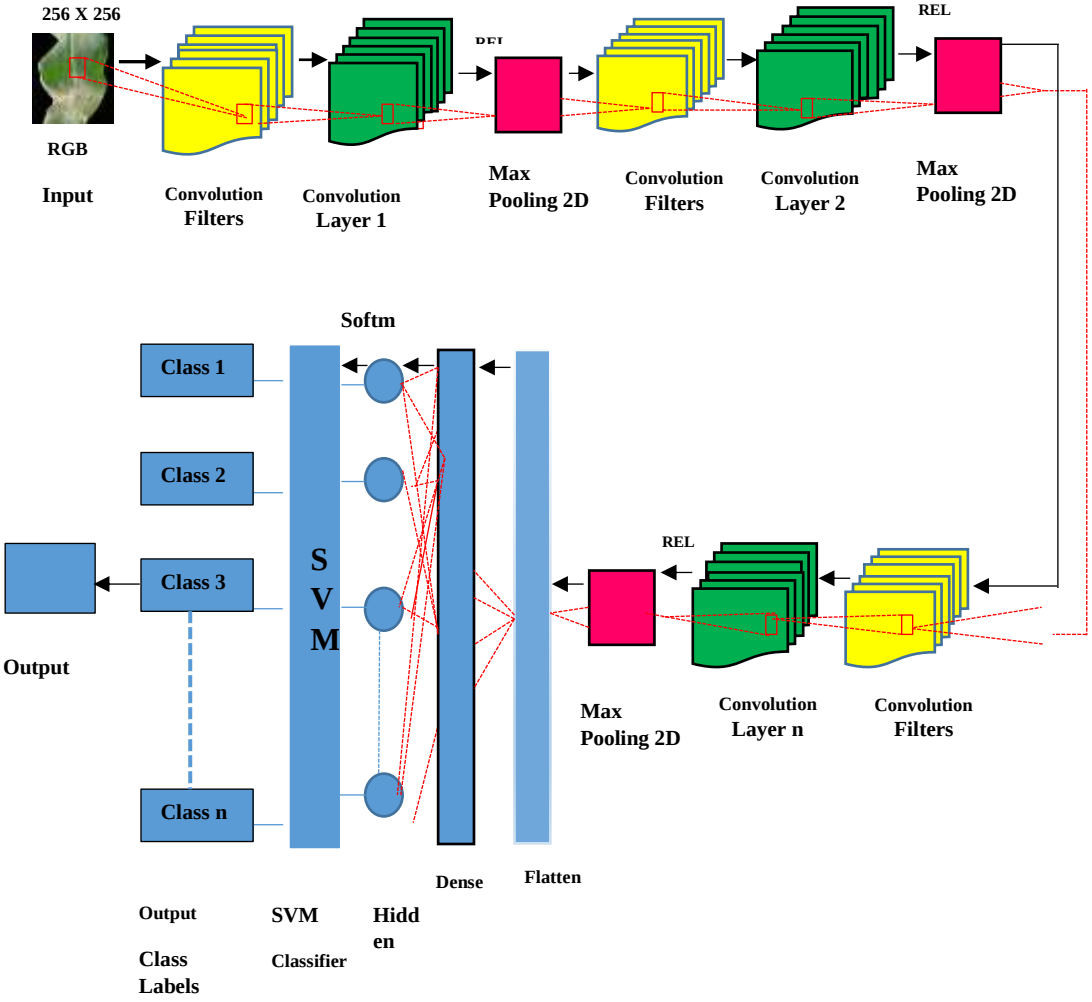


Figure 5.1 CNN-SVM Model

5.3 Convolution Layer

A hybrid model of a deep neural network that used Convolutional Neural Network (CNN) and Support Vector Machine (SVMs) to identify and classify crop diseases and pests relied heavily on the convolutional layer. In order for the model to learn spatial hierarchies and important features like edges, textures, and patterns, this layer was vital in feature extraction by applying a sequence of filters to the input images. The input image was convolved using a set of learnable filters or kernels in each convolutional layer to generate feature maps. It was mathematically possible to express the convolution procedure as;

$$y_{i,\,j,\,k} = \sum_{m,n} x_{i+m,\,j+n,\,k} . w_{m,\,n,\,k} + b_k \qquad \text{(Eq. 5.1)}$$

where the output feature map was denoted as $y_{i,j,k}$, the input image was denoted as x, the convolution kernel was denoted as w, and the bias term for the k-th filter was denoted as b_k. The feature map's spatial position was represented by the indices i and j, and the filter's dimensions were represented as m and n.

The convolutional layers captured data at different abstraction levels; the first layers captured simple features like edges and textures, while the last layers captured more intricate patterns and forms that indicated pests or crop diseases. To differentiate between healthy and affected crops and to identify different types of pests, this hierarchical feature extraction was important.

The convolutional layer used padding, which maintained the input's spatial dimensions, to improve the model's feature extraction. Because of this, the filters were able to extract features from the margins of the image while covering the full image. To guarantee accurate feature extraction, it was common practice to set the stride to one, which controlled the step size of the filter throughout the image.

Also, to introduce non-linearity, the convolutional layer used the Rectified Linear Unit (ReLU) activation function. The ReLU activation function was defined as;

$$f(x) = \max(0, x) \quad \text{(Eq. 5.2)}$$

The network was able to accurately identify various crop diseases and pests because of its non-linearity, which enabled it to understand intricate patterns and interactions within the images [258].

The enhanced deep neural network model's convolutional layer played a critical role in converting unstructured image input into a structured representation of features that the Support Vector Machine (SVM) component could subsequently classify. Because of this technique, the model reliably and accurately aided in decision-making in crop diseases and pest management.

5.4 Activation Function

An important part of the hybrid deep neural network model's activation function was adding non-linearity; this allowed the model to correctly identify and classify crop diseases and pests using a combination of CNN and SVM. To learn and represent

complex patterns and connections in the data, the model relied on this non-linearity. This was important for recognising pests and differentiating between healthy and affected crops. The Rectified Linear Unit (ReLU) was the activation function that was used most often in this model.

The researcher defined the ReLU activation function as;

$$f(x) = \max(0, x) \qquad \text{(Eq. 5.3)}$$

The input was changed using this simple function, which kept positive numbers and set negative values to zero. Among ReLU's many benefits was its ability to speed up training convergence by reducing the impact of the vanishing gradient problem, a common problem with older activation functions such as sigmoid and tanh. By reducing gradients during backpropagation, the vanishing gradient problem made it difficult to train the deep neural network, as the model had a harder time updating its weights effectively. To solve this problem, ReLU made sure that the gradient for positive inputs was constant, which allowed for faster weight updates.

The convolutional layers were able to take crop and pest images and learned intricate features and hierarchical patterns due to the nonlinearity introduced by the ReLU activation function. To properly classify various crop diseases and pests, these extracted important features. To solve the vanishing gradient problem, where neurons became inactive if the input was continuously negative, the researcher used variations of ReLU, including Leaky ReLU and Parametric ReLU (PReLU). For instance, for negative values, Leaky ReLU created a slight slope;

$$f(x) = \begin{cases} x \text{ if } x > 0 \\ \alpha x \text{ otherwise} \end{cases} \qquad \text{(Eq. 5.4)}$$

in which the modest constant α was present. The ability of the neurons to keep learning in the face of negative inputs was guaranteed by this variation.

To convert the convolutional layers' linear outputs into non-linear representations of distinct features, the activation function component was important. The Support Vector Machine classifier was able to correctly classify images by using the extracted features to define decision boundaries. The activation function component greatly enhanced the model's ability to accurately support decision-making in crop diseases

and pest management by facilitating the extraction and processing of complex features [258].

5.5 Pooling Layer

The combination of the Convolutional Neural Network (CNN) with the Support Vector Machine (SVM), the hybrid deep neural network model was able to reduce the spatial dimensions of the feature maps while retaining important information. This allowed for accurate identification and classification of crop diseases and pests. When working with large datasets in agricultural applications, reducing dimensionality was important since it reduced computational complexity and mitigated the overfitting problem. Max pooling was the pooling procedure that was used most often in this hybrid model

When using max pooling, a 2x2 window was slid over the input feature map, and the largest value inside each window was selected. This process successfully reduced the feature map to its essential features while eliminating redundancy. Max pooling was mathematically represented as;

$$y_{i,\,j,\,k} = \max_{m,n}(x_{i+m,\,j+n,\,k}) \qquad \text{(Eq. 5.5)}$$

$x_{i+m,\,j+n,\,k}$ denoted the input feature map values inside the pooling window, while $y_{i,j,k}$ denoted the output of the pooling layer.

The spatial size of the feature map was reduced using max pooling, which focused on the most important features. As a consequence, the succeeding layers required fewer parameters and calculations. This reduction was crucial for keeping the model size reasonable and enhancing computing performance, particularly when processing crop leaf diseases and pest images with high quality.

The model implementation also made use of average pooling, which was an alternative to max pooling that calculated the mean of all pooling windows.

$$y_{i,\,j,\,k} = \frac{1}{N}\sum_{m,n} x_{i+m,\,j+n,\,k} \qquad \text{(Eq. 5.6)}$$

In which the number of values in the pooling window was denoted as N. Although max pooling extracted distinct features better, average pooling gave a smoother representation by averaging all the values.

Additionally, the pooling layer introduced translational invariance, which meant that the pooled features were mostly unaffected by minor changes or distortions in the input images. When used in crop disease and pest classification, this invariance proved useful since crop leaf diseases and pest images varied depending on factors like illumination, orientation, and camera angle.

The pooling layer played a crucial role in the hybrid deep neural network model by reducing high-dimensional features to low dimensions. As a result, the Support Vector Machine classifier was able to identify the most important features for determining whether crops were healthy or affected and for classifying pests. The model's ability to efficiently and robustly provide decision support for crop diseases and pest management was enhanced by the pooling layer, which successfully managed the spatial dimensions and computational complexity.

5.6 Fully Connected Layer

The final step before classification was the hybrid deep neural network model's fully connected layer. This layer integrated CNN and SVM to correctly identify and classify crop diseases and pests. The convolution and pooling layer extracted high-level features; this layer aggregated and transformed the extracted features into a format that the Support Vector Machine (SVM) classifier could use. The neurons in the fully connected layer integrated data from the whole input image since they were connected to every neuron in the preceding layer.

A fully connected layer's mathematical representation of its operation was;

$$y_i = f\left(\sum_j w_{ij} x_j + b_i\right) \qquad \text{(Eq. 5.7)}$$

y_i was the output of the i-th neuron, w_{ij} was the weight linking the j-th input to the i-th neuron, x_j was the j-th input, and b_i was the bias term. In most cases, the network was

able to understand intricate patterns and correlations in the data by using a non-linear activation function known as ReLU.

In order to detect complex patterns that were signs of pests or crop diseases, the network's fully connected layer combined features from regions of the image. Important for the correct classification of complex and diverse crop diseases and pest images, this layer allowed the model to draw global conclusions about the whole image. The final product of the fully connected layer was a feature vector that captured the input image's most important and unique features. The final classification was performed by passing this feature vector into the Support Vector Machine classifier. Decision boundaries were established by the SVM by using high-dimensional feature representation to differentiate between various types of pests and crop diseases.

The hybrid model's ability to convert CNN-learned spatial hierarchies into a structured format usable by SVM for high classification accuracy was made possible by the incorporation of the fully connected layer. The model's ability to provide reliable decision support in crop diseases and pest management, assisting farmers and agronomists in intervening quickly and effectively, was dependent on the smooth integration of feature extraction and classification components.

5.7 Feature Vector Generation

The learned features had to be transformed into a structured format that could be used for classification by the feature vector generation component of the hybrid deep neural network model. This component merged Convolution Neural Network (CNN) and Support Vector Machine (SVM) to correctly identify and classify crop diseases and pests. This part was the link between the feature extraction and classification processes in the SVM; it came after the CNN's fully connected layer. This component's principal objective was to concisely and informatively extract the input images' most relevant and distinctive features.

The feature vector, a one-dimensional vector, was created by flattening the high-dimensional feature maps after the input images were processed by the final pooling and fully connected layer. To prepare the data for classification with the SVM classifier, this transformation required transforming the spatially organised data into a

linear format. The process of generating feature vectors was mathematically represented as;

$$v = [x_1, x_{2,\ldots,} x_n] \quad \text{(Eq. 5.8)}$$

where the feature vector was denoted by v and the individual feature values obtained from the last CNN layer as x_i. This vector had all the important features that were learned by the convolutional layer; it captured all the important elements of the image, including patterns, textures, and forms that were signs of pests or crop diseases. To make sure the SVM had fully summarised input for classification, the feature vector efficiently summarised the hierarchical representations learned by the Convolution Neural Network. Because it included the most distinctive features needed to differentiate between various types of crop diseases and pests, this representation improved the SVM's ability to identify decision boundaries.

Additionally, in order to improve computing efficiency and reduce duplication, it was common practice to reduce the feature vector's dimensionality using methods like dimensionality reduction and feature selection. This reduction enhanced the model's accuracy and resilience by directing the classification effort toward the most distinctive features.

In the hybrid deep neural network model, the feature vector-generating component played a crucial role in integrating CNN and SVM by providing a structured and useful representation of the input data. By combining CNN and SVM, the model was able to make use of their respective strengths, creating a decision support system for crop diseases and pest management that is both accurate and reliable [259].

5.8 Support Vector Machine Classifier

The hybrid deep neural network model's Support Vector Machine (SVM) classifier was important in correctly identifying and classifying pests and crop diseases. The model used the Convolutional Neural Network's (CNN) robust feature extraction capabilities to generate a feature vector that the Support Vector Machine used as input. With its strength in high-dimensional data and ability to define appropriate decision boundaries, the SVM was responsible for the classification task.

Support Vector Machine (SVM) was well-known for its hyperplane component capabilities, which allowed it to optimise the margin between feature space classes. In this hybrid model, the SVM identified these decision boundaries and separated different classes of crop diseases and pests using the feature vectors created by CNN. The mathematical definition of the decision function of a Support Vector Machine was;

$$f(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b\right) \quad \text{(Eq. 5.9)}$$

where the Lagrange multipliers were denoted as α_i , the class labels were denoted as y_i, the kernel function was denoted as $K(x_i, x)$, and the bias term was denoted as b. In order to make the features separable, the kernel function, which was a radial basis function (RBF), converted the features into a higher-dimensional space.

$$K(x_i, x) = \exp(-\gamma \| x_i - x \|^2) \quad \text{(Eq. 5.10)}$$

This kernel technique allowed the SVM to deal with non-linear, complex correlations in the data; this helped to distinguish between pests and crop diseases that were slightly different from one another hence improving the classification accuracy.

The SVM's goal during training was to locate the hyperplane that would allow the margin between the classes to be maximised. This required finding the values of α_I and b, which established the decision boundary, by solving a quadratic optimisation problem. The goal of the Support Vector Machine (SVM) was to maximise the margin between classes while minimising the classification error.

To complement the CNN's feature extraction capabilities, a strong classification mechanism was added to the hybrid deep neural network model by integrating the Support Vector Machine. The SVM achieved great classification accuracy, accurately classifying different crop diseases and pests, by using the most distinctive features extracted by the CNN. Using SVM as the final classifier in the model improved its capacity to generalise, enabling it to perform well on unseen data.

In general, the SVM classifier played an important role in the hybrid deep neural network model's effectiveness, offering a reliable decision support system for crop diseases and pests. Timely and effective interventions to protect crops and guarantee sustainable farming practices were made possible by the model, which used the

complementary features of CNN and SVM as reliable tools for precision agriculture [260].

5.9 Training the Hybrid Model

To train the hybrid deep neural network model, a combination of Convolution Neural Network (CNN) and Support Vector Machine (SVM) to correctly identify and classify crop diseases and pests, a multi-stage procedure was employed to optimise the performance of the SVM in classification and the CNN in feature extraction. The goal of the training was to create a model with good data generalisation capabilities and reliable decision support for crop diseases and pest management.

The Convolution Neural Network (CNN) part of the model was the first to undergo training; it was responsible for extracting features from the crop diseases and pests input images. The Convolution Neural Network (CNN) was trained on a labelled dataset of crop diseases and pest images, where each image was assigned the appropriate pest or crop disease category. By using backpropagation and Stochastic Gradient Descent (SGD), the goal of the training process was to optimise the network's biases and weights in order to reduce the classification error. The Convolutional Neural Network (CNN) training loss function used the categorical cross-entropy loss, which was;

$$L(y, \hat{y}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \qquad \text{(Eq. 5.11)}$$

$\hat{y}$ denoted the predicted probability distribution, C stood for the number of classes, and y denoted the actual class labels. To direct the network's learning, this loss function measured the difference between the actual and predicted probability distributions.

The output of the fully connected layer was transformed into a feature vector, which extracted the most distinctive features of the input images, as the CNN learned to extract hierarchical features. The Support Vector Machine (SVM), as part of the model, used these feature vectors as inputs.

To determine the best decision boundary for classifying crop diseases and pests, the SVM was trained using the feature vectors extracted by the CNN. To optimise the margin between classes, as described by the support vectors, the SVM had to be

trained to solve a quadratic optimisation problem. Here was the formulation of the optimisation problem;

$$\min_{\alpha}\left(\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{N}\alpha_i\right) \quad \text{(Eq. 5.12)}$$

In light of the limitations:

$$0 \le \alpha_i \le C, \sum_{i=1}^{N}\alpha_i y_i = 0 \quad \text{(Eq. 5.13)}$$

Where the Lagrange multipliers were denoted by α_i, the regularisation parameter was denoted by C, the class labels were denoted by y_i, and the kernel function was denoted as $K(x_i, x_j)$. The SVM was able to efficiently manage non-linear decision boundaries because of the kernel function, which was often used as a Radial Basis Function (RBF).

Iterative training was used to increase the overall performance of the hybrid model by simultaneously refining the CNN and SVM components. The use of a transfer learning method was used, with the Convolutional Neural Network (CNN) first trained on a large dataset and then fine-tuned on the crop diseases and pest dataset in question. Improved model performance for crop diseases and pest classification was achieved by making use of the model's pre-learned features from standard images.

To guarantee that the trained model could generalise well to new data, it was tested on a validation set. To evaluate the efficacy of the model in practical settings, performance measures including recall, accuracy, precision, and F1 score were used.

In order to train the hybrid model, the researcher optimised the Convolution Neural Network (CNN) and Support Vector Machine (SVM) layers to build a robust and reliable decision-support system for crop diseases and pest management. The improved model resulted in accurate classifications and proper recommendations for precision agriculture by integrating a Convolution Neural Network with a Support Vector Machine [261].

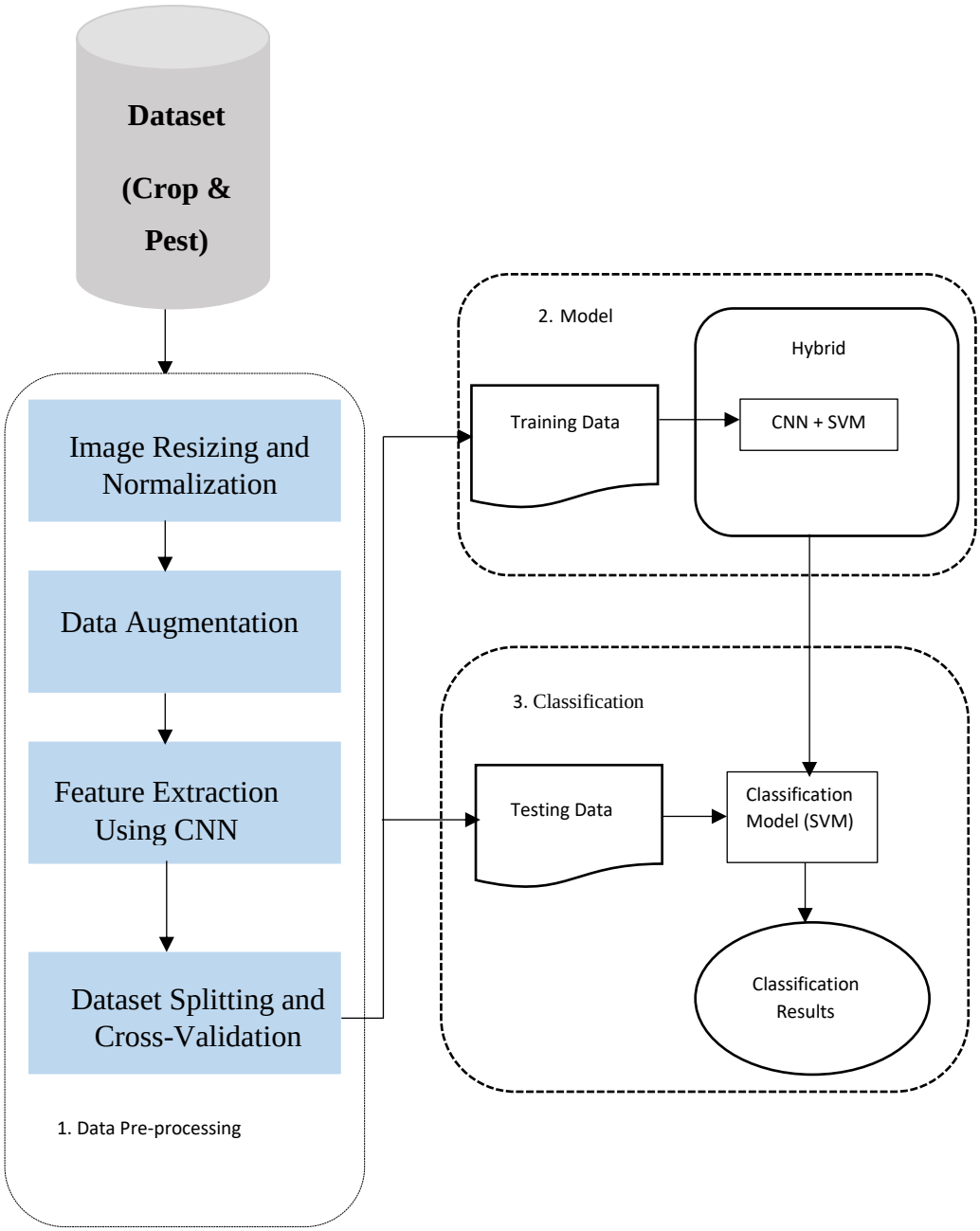


Figure 5.2 Schematic Diagram for Crop Diseases and Pests Classification Model

Convolutional Neural Network (CNN) and Support Vector Machine (SVM) worked together to build a hybrid deep neural network model that accurately identified and classified crop diseases and pests. The model's interdependent parts made it easy to analyse, learn, and classify images of crop diseases and pests, as shown in Figure 5.2. The success of the classification and its robustness in different environments depended on each step.

The Dataset Component was the first step; it included images of crop diseases and pests. A wide range of crop diseases and pests were represented in the dataset's samples. The generalizability and accuracy of the model were greatly affected by the dataset's variety and quality. To avoid class imbalance problems that could skew the classification results, the dataset was designed to include an equal number of crop diseases and pest images.

The input data was standardised through the use of image resizing and normalisation. To ensure that the CNN model could handle images with varying dimensions, all of the images were reduced to a standard dimension of 224 × 224 pixels. The normalisation process used the following formula to convert pixel values to a range of [0,1];

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad \text{(Eq. 5.14)}$$

This made sure that the model learned well regardless of the image's brightness or illumination.

Data Augmentation was used to make the model more robust and prevent it from overfitting. The researcher rotated, flipped, zoomed, cropped, and adjusted the contrast on the images, among other adjustments. The model was able to generalise more accurately with the use of augmented images since they mimicked real-world data variances. A mathematical formula was used to calculate the augmented image I ′;

$$I' = T(I) \quad \text{(Eq. 5.15)}$$

where T represented the transformation function applied to the original image I.

The convolutional layers extracted useful information from the input images by using a CNN through the feature extraction process. Activation functions like Rectified Linear Unit (ReLU) and pooling layers were used to reduce the dimensionality after convolutional layers in the CNN. The representation of the convolution operation was;

$$F(i,j)=\sum_m \sum_n I(i+m,j+n).K(m,n) \quad \text{(Eq. 5.16)}$$

where I(i,j) was the input image,K(m,n) was the convolutional kernel, and F(i,j) was the feature map.

Differentiating between affected and healthy crops relied on the spatial hierarchies of features extracted by CNN. These features included edges, textures, and patterns.

Dataset splitting and cross-validation were performed after feature extraction, enabling reliable model assessment. Approximately 70% of the dataset was used during training, 15% in validation, and 15% in testing. The model's performance was validated across subsets of data using K-fold cross-validation. The total accuracy for k-fold validation was calculated as;

$$A = \frac{1}{k}\sum_{i=1}^{k} A_i \quad \text{(Eq. 5.17)}$$

where A_i represented the accuracy obtained in the i-th fold. This method enhanced the model's robustness and prevented it from overfitting.

The Support Vector Machine was used for classification, whereas the CNN acted as a feature extractor. The features extracted by the CNN were fed into a Support Vector Machine classifier rather than a fully connected layer. The definition of the SVM decision function was represented as;

$$f(x)=w\cdot\phi(x)+b \quad \text{(Eq. 5.18)}$$

where w was the weight vector, $\phi(x)$ represented the feature transformation function, and b was the bias term.

The SVM used the learned decision boundaries to classify the input as either a crop disease or a pest.

Model Training included Convolutional Neural Network (CNN) layers learning feature representations and Support Vector Machine (SVM) layers optimising the hyperplane component to clearly separate classes. While training CNNs, the cross-entropy loss function was used.

$$L = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \quad \text{(Eq. 5.19)}$$

where y_i was the actual label, and $\hat{y}_i$ was the predicted probability.

The SVM, in contrast, reduced hinge loss to a minimum value.

$$L_{SVM} = \sum_{i=1}^{N} \max(0, 1 - y_i f(x_i)) \quad \text{(Eq. 5.20)}$$

It made sure the margins for classification were maximised.

Final predictions were made using the SVM classification model using the features extracted by the CNN. Healthy crops, common rust, leaf spots, and pests were all classified by the SVM according to the hyperplane that maximised the margin between classes. To improve non-linearly separable data, the kernel technique was used with the function.

$$K(x_i, x_j) = \phi(x_i).\phi(x_j) \quad \text{(Eq. 5.21)}$$

where K represented the kernel function, mapping the input data into higher-dimensional space for better classification.

Stakeholders were able to make better decisions and intervene faster in crop disease and pest management because of the model's accurate predictions. This technique was applied to real-world agricultural applications because it combined a Convolutional Neural Network (CNN) for feature extraction with a Support Vector Machine (SVM) for classification, which greatly improved accuracy, robustness, and generalizability [262].

Algorithm: Hybrid CNN-SVM Model for Crop Diseases and Pest Classification

Input: Image dataset of crops with labeled disease and pest classes
Output: Trained hybrid CNN-SVM model for classification
Procedure:

1. **Data Collection and Preprocessing**
   (a) Collect a dataset of crop images labeled with various crop diseases and pests.
   (b) Preprocess images; resize, normalize, and apply data augmentation for variability.
2. **CNN Model Training**
   (a) Initialize CNN with multiple convolutional and pooling layers.
   (b) Train CNN on the preprocessed dataset.
   (c) Optimize CNN parameters, e.g., number of layers, filter size, learning rate, to maximize accuracy.
   (d) Extract features from the final fully connected layer of CNN.
3. **SVM Classifier Training**
   (a) Use CNN-extracted features as input to an SVM classifier.
   (b) Choose the RBF kernel and tune parameters (gamma, C) through cross-validation.
   (c) Train SVM to classify extracted features into crop disease or pest classes.
4. **Evaluation and Comparison**
   (a) Evaluate the hybrid CNN-SVM model on the test set.
   (b) Measure performance metrics: accuracy, precision, recall, F1-score.
   (c) Compare results to standalone CNN and other models (e.g., DenseNet, VGG) for validation.
5. **Output Final Model**

Figure 5.3 CNN-SVM Algorithm

To classify crop diseases and pests, the researcher developed a hybrid CNN-SVM model that used a Convolution Neural Network for feature extraction and a Support Vector Machine for classification. To guarantee the correct classification of crop diseases and pests, the algorithm adhered to a methodical procedure.

Collecting image datasets, including different types of crop diseases and pests, was the initial stage in the data gathering and preparation procedure. To guarantee accurate supervised learning, each image was labelled. The images were then resized to a consistent size of 224×224 pixel resolution to prepare them for preprocessing by the CNN. Pixel values were scaled between 0 and 1 through a normalisation technique, which improved convergence during model training. Rotating, flipping, and zooming were some of the data augmentation methods used to increase variability and avoid overfitting. To create a testing subset of the dataset and a training subset of the dataset, the dataset was divided into two sets.

The CNN model was trained using the preprocessed dataset after the completion of preprocessing. The CNN design had convolutional layers that used 3×3 filters, and then ReLU activation functions were used to include non-linearity. To reduce spatial dimensions while retaining distinct features, pooling layers were used. The model was

refined using pooling and convolutional layers before being flattened into a dense layer for feature extraction. For multi-class classification, the CNN model was built using the Adam optimiser with categorical cross-entropy as the loss function. Accuracy was maximised during training by optimising parameters including learning rate, number of layers, and filter sizes. Ten iterations of training with a batch size of thirty-two were used to train the model, and twenty per cent of the training data was used for validation.

The CNN-extracted features were fed into the SVM classifier after training. The CNN extracted features from input images and turned them into numerical features, which were passed to the SVM for classification. The SVM classifier, trained with a Radial Basis Function kernel to deal with non-linearly separable data, was then fed with the features extracted by the CNN. To enhance the classification accuracy, the SVM parameters, such as gamma and C, were tuned during cross-validation.

The hybrid model was tested and compared with different deep learning architectures, including DenseNet, AlexNet, and VGG, as well as a standalone CNN model. Precision, accuracy, recall, and F1-score were some of the important metrics used to evaluate performance. The researcher compared the SVM classifier's predictions to the actual truth labels after subjecting it to the unseen test dataset. Because Support Vector Machine (SVM) optimised decision boundaries across various pest and crop disease categories, the hybrid CNN-SVM model outperformed other deep learning architectures, which included DenseNet, AlexNet, and VGG, as well as the standalone CNN model.

The SVM's generalizability was checked using k-fold cross-validation, which was done to test the model's robustness. When the Convolutional Neural Network (CNN) model failed to correctly identify crop diseases and pests because of overfitting and complex crop disease and pest patterns, the hybrid model produced more accurate classification results.

The trained hybrid model was integrated into the software application, guaranteeing its incorporation into the decision-support system for crop disease and pest identification. Farmers and agronomists were able to upload crop disease and pest

images into the web application integrated with the model and got accurate classification results.

Improvement in the identification and classification of crop diseases and pests was achieved by combining the feature extraction capabilities of CNN with the classification ability of SVM in a hybrid CNN-SVM model. The model offered a scalable and effective solution for precision agriculture by using both deep learning and machine learning algorithms [263].

5.10 Model Evaluation

An important part of making sure the hybrid deep neural network model worked well in the real-world scenario was testing it. The model used a Convolution Neural Network (CNN) and a Support Vector Machine (SVM) to correctly identify and classify crop diseases and pests. To provide accurate decision support for crop diseases and pest management, it was important to evaluate the model's ability to generalise to unseen data.

Accuracy, precision, recall, and F1 score were among the performance metrics used to evaluate the model. The metrics used showed how well the model could classify images into their respective classes.

The total correctness of the model was measured by accuracy, which was defined as;

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad \text{(Eq. 5.22)}$$

The precision was determined by dividing the number of correct predictions by the total number of positive predictions produced by the model and was expressed as;

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad \text{(Eq. 5.23)}$$

The ability of the model to accurately identify all positive cases was assessed by Recall (Sensitivity), which was stated as;

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad \text{(Eq. 5.24)}$$

As the harmonic mean of recall and accuracy, the F1 Score provided a balanced measure of the model's performance;

$$\mathrm{F1Score} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \qquad \text{(Eq. 5.25)}$$

To avoid unfairly placing too much emphasis on recall or accuracy when testing models on unbalanced datasets, the F1 score proved helpful.

A confusion matrix, showing the total number of true negatives, false positives, false negatives, and true negatives for each category, was also generated as part of the assessment process to highlight how well the model performed across all classes. Using this matrix, the researcher was able to identify potential problem areas with the model, such as the most often misclassified classes.

The model's performance was further assessed using Area Under the Curve (AUC) and Receiver Operating Characteristic (ROC) curves for various threshold values, providing a more detailed picture of its prediction ability.

Modifications to the model's architecture, Support Vector Machine (SVM) parameters, and feature extraction procedures were all influenced by the evaluation results, which led to model improvement and fine-tuning. The researcher improved decision-making for agronomists and farmers managing crop diseases and pests by systematically testing the model to guarantee accurate and reliable classifications. To ensure the model's applicability in various agricultural conditions and to create confidence in its predictions, systematic validation was needed [264].

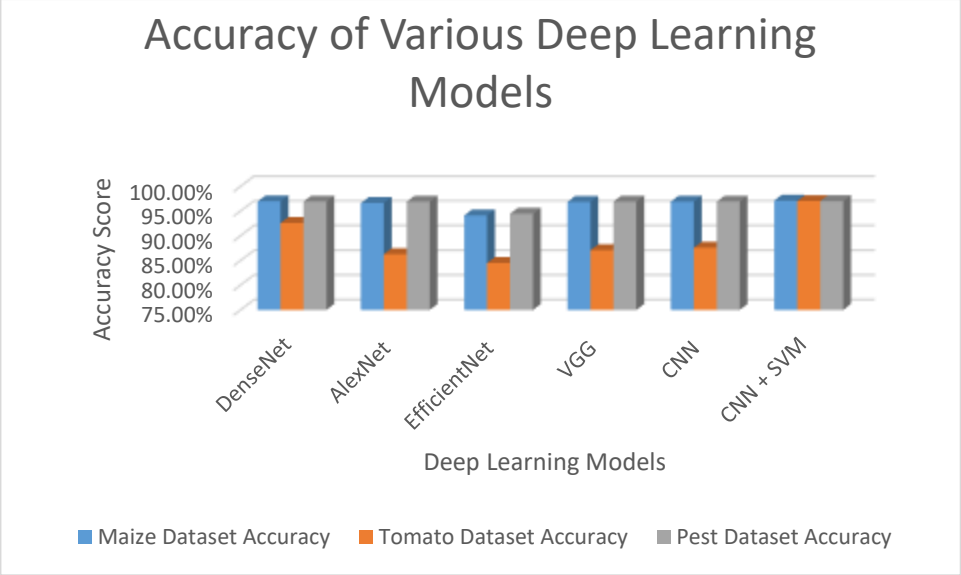


Figure 5.4 Accuracy of Various Deep Learning Models

Figure 5.4 shows the accuracy performance of six deep learning models on the pest, maize, and tomato datasets. In all three datasets, the CNN+SVM hybrid model outperformed the other models, proving its effectiveness in crop disease and pest classification. In particular, when compared to the other models, the CNN+SVM model had the highest accuracy rates for maize leaf diseases (97.13%), tomato leaf diseases (97.04%), and pest classification (97.02%). Decision boundaries were improved, and misclassification errors were reduced as a result of combining the Convolution Neural Network for feature extraction with the Support Vector Machine for classification.

With an accuracy of 96.99% for maize leaf diseases, 92.60% for tomato leaf diseases, and 96.94% for pest classification, DenseNet demonstrated comparable performance among the other deep-learning models. According to these results, DenseNet was well-suited for complex image classification tasks because of its dense connection structure, which efficiently retained feature representations. On the other hand, when compared with the hybrid CNN-SVM model, its poor performance on the tomato dataset indicated that it had difficulty in generalising features.

DenseNet and the hybrid model had higher accuracies than AlexNet and EfficientNet. When it came to the maize data set, AlexNet had an accuracy of 96.66% on the maize leaf diseases data set, 86.30% on the tomato leaf diseases, and 96.91% on pest

classification; in comparison, EfficientNet got 94.14%, 84.50%, and 94.47% respectively. Although EfficientNet's architectural design was geared toward computational efficiency, its poor performance when subjected to crop leaf disease datasets suggested that it missed critical crop disease-related features. Because of its older architecture, AlexNet had a generalisation problem with the tomato leaf diseases dataset and performed worse than the other models.

For the maize dataset, VGG achieved 96.88% accuracy, for tomato 87.10%, and pest classification 96.93%. The CNN achieved 87.60% on the tomato dataset, 96.93% for maize, and 96.94% for pests. Based on these results, the VGG's deeper design allowed for efficient feature learning. It was found that a well-optimised CNN architecture provided good classification results without the complexity of deeper networks, since the conventional CNN model performed marginally better than VGG, especially with the tomato dataset.

The hybrid CNN+SVM model proved to be the most robust in crop leaf diseases and pest classification, constantly outperforming all other models. An improvement in accuracy across all datasets was achieved by combining CNN's hierarchical feature extraction with SVM's effective classification technique. Based on the findings, the CNN+SVM model was a better choice for classifying crop leaf diseases and pests in agriculture since it used hybrid deep learning techniques to increase image classification accuracy [265].

The developed hybrid CNN-SVM model was optimised to maximise the computational efficiency while maintaining the accuracy of crop pest and disease classification. This facilitated the model to work effectively on resource-limited devices and, thus, made it more relevant to actual agricultural situations where internet, memory, and processing capacity were limited.

5.11 Summary

To support the identification and classification of crop leaf diseases and pests, a hybrid deep neural network model was developed. This model used a hybrid of Convolutional Neural Network (CNN) and Support Vector Machine (SVM), which helped in decision support for crop diseases and pest management. This hybrid model combined

the strength of a Convolution Neural Network (CNN) with a Support Vector Machine (SVM), making it an effective tool for precision agriculture compared to other models.

The Convolution Neural Network was responsible for handling high-resolution crop diseases and pest images, with the goal of extracting complex features. The Pooling layer reduced spatial dimensions, retaining the most important features, whereas the convolution layer extracted local patterns like textures and edges. The network was able to learn complex patterns that assisted in the classification of crop leaf diseases and pests.

In order to describe the overall qualities of each image, the CNN's fully connected layer combined all of the extracted features into a full feature vector. Using its capability to identify optimum decision boundaries in high-dimensional spaces, the SVM classifier correctly classified the images into their respective classes of crop leaf diseases and pests. When dealing with crop leaf diseases and pests data, which naturally has complexity and unpredictability and often has slight distinctions across classes, the SVM component proved to be the most accurate in classifying the extracted features. Training the CNN on labelled datasets improved its feature extraction abilities and laid the step for the subsequent training of the hybrid model.

Accuracy, precision, recall, and F1 score were some of the metrics used in the model validation process to guarantee model performance on unseen data. To ensure the model accurately classified images in actual agricultural contexts, this assessment was important for confirming its generalizability.

An invaluable tool for early diagnosis and control of crop leaf diseases and pests, the hybrid deep neural network model outperformed other models in terms of classification accuracy. The model provided a state-of-the-art solution for decision support in precision agriculture by combining CNN and SVM techniques. This made it to efficiently bridge the gap between deep learning's capabilities in feature extraction and SVM's classification ability. As a result of this integration, farmers and the agricultural sector as a whole benefited from improved crop health and increased productivity.

CHAPTER SIX
MODEL IMPLEMENTATION AND APPLICATION INTEGRATION

6.1 Introduction

To guarantee that the application could handle the demanding requirements of precision agriculture, great precaution was taken during the model's implementation. This model made use of a Convolution Neural Network architecture and a Support Vector Machine to provide timely and practical decision support. Starting with the definition of key criteria, the implementation process addressed the complex and non-linear data relationships inherent in crop diseases and pest control, as well as the need for high accuracy and quick response times. The researcher selected a deep learning approach because it automatically extracted distinct features from large datasets without requiring a lot of human intervention. The selected architecture was a multi-layer neural network with input layers tailored to handle a wide range of crop diseases and pest images. To make sure the model could successfully capture complex patterns in the images, especially when it came to identifying early symptoms of crop diseases and pests, the hidden layers were fine-tuned through hyperparameter optimisation.

Various techniques were used during implementation to avoid overfitting, such as cross-validation and dropout regularisation, to improve the model's generalisation performance across crop leaf diseases and pest datasets. To make sure the model could adjust to different weather circumstances, it was trained using a massive dataset obtained from the Kaggle website. To minimise the loss function and improve the classification accuracy over time, the network's weights were repeatedly updated using the backpropagation technique. Since timely outputs were important for real-time decision-making in crop leaf diseases and pests management, where delays could result in substantial crop losses, this backpropagation technique was considered.

The model was incorporated into a decision-support system that offered farmers and agricultural extension officials practical insights. The results were presented in a way that even people without technical knowledge could understand, with visual dashboards and alert systems included, so that they could act quickly on the recommendations. In the deployment phase, the model was tested in real-world settings, where it was assessed for its performance under many conditions, such as various types of crop diseases and different types of pests. Scalability and flexibility

were crucial throughout implementation to guarantee the model could be readily updated, extended to include new data sources, or adjusted to accommodate future technology developments. In the end, stakeholders were able to reduce crop losses and increase yield outcomes with the help of this model based on Convolution Neural Networks and Support Vector Machine, which greatly improved the efficiency and timeliness of decision support in crop diseases and pest management [266].

6.2 Deployment Strategy

The researcher used a well-structured deployment process to implement the hybrid model of Convolution Neural Network (CNN) and Support Vector Machine (SVM) to use in a real-time decision-support application. One of the steps following CNN-SVM model training was to serialise the trained model using Pickle, which was a model library used to easily save and load the model without retraining. The next step was to move the model to the cloud, where it could make use of resources like Digital Ocean, which allowed for scalability and real-time performance.

The HTML, JavaScript, and CSS languages were used for front-end development, which provided a user-friendly interface for stakeholders like decision-makers, agronomists, and farmers to interact with the model. Through this interface, users uploaded images of crop leaf diseases and pests. The model extracted features from the uploaded inputs using a Convolutional Neural Network (CNN), and the model then identified significant patterns or abnormalities in the images that were associated with pests or crop diseases.

After feature extraction, the SVM component performed the final classification, identifying pests or crop diseases that may have affected the crop and offering insights into potential causes. Deploying the model using the FastAPI framework improved the real-time classification by linking the back-end model to the front-end interface, allowing users to receive the classification results faster.

The model was integrated with a load balancer to efficiently handle requests and provide timely responses despite the huge amount of incoming data from different users. The dashboard thereafter showed real-time predictions, which provided in-depth insights and recommendations like crop disease identification, pest management, and prevention strategies. The goal of these recommendations was to

help those involved in crop health management and crop disease and pest prevention make better decisions [267].

6.3 Integration with the Application

This section explains how the web application and the hybrid CNN and SVM model worked together to provide batch and real-time decisions about pest and disease management in crops. Beginning with the model's development, the process included combining a Convolutional Neural Network (CNN) for robust feature extraction and a Support Vector Machine (SVM) for accurate classification. To make sure this hybrid model was accurate and robust, it was trained and validated on the correct datasets.

The front end of the web application was developed using CSS, JavaScript, and HTML. The web pages were structured using HTML, interactive components and asynchronous data processing were handled by JavaScript, and the application's interface was styled using CSS to make it both visually attractive and user-friendly. Users were able to upload images and see results easily because of the dynamic and responsive user experience generated by these technologies.

The researcher used FastAPI to build the application's backend because of how quickly and efficiently it processed web requests and responses. To analyse incoming data and provide predictions, FastAPI communicated directly with the hybrid model. The RESTful API endpoints enabled the front end to interact with the backend through HTTP requests, thereby facilitating seamless communication.

Whether users were inputting data in batch or real-time, the web application would send the requests to the FastAPI backend over HTTP. The FastAPI then sent the information to the hybrid CNN-SVM model. To aid in classification and decision-making, the CNN component extracted distinct features from the images and fed them into the SVM for classification. Next, the FastAPI backend received the model's output, which included predictions and recommendations.

When it came to making decisions in real-time, the FastAPI backend handled the data and predictions immediately, so users received feedback immediately. With batch processing, the backend could handle massive amounts of images, either in batch or through scheduled operations, and then send the results to the front end to be displayed.

By integrating state-of-the-art deep learning techniques with efficient web technologies, this integration guaranteed that those involved in crop disease and pest management had access to quick and accurate decision support through a seamless web application.

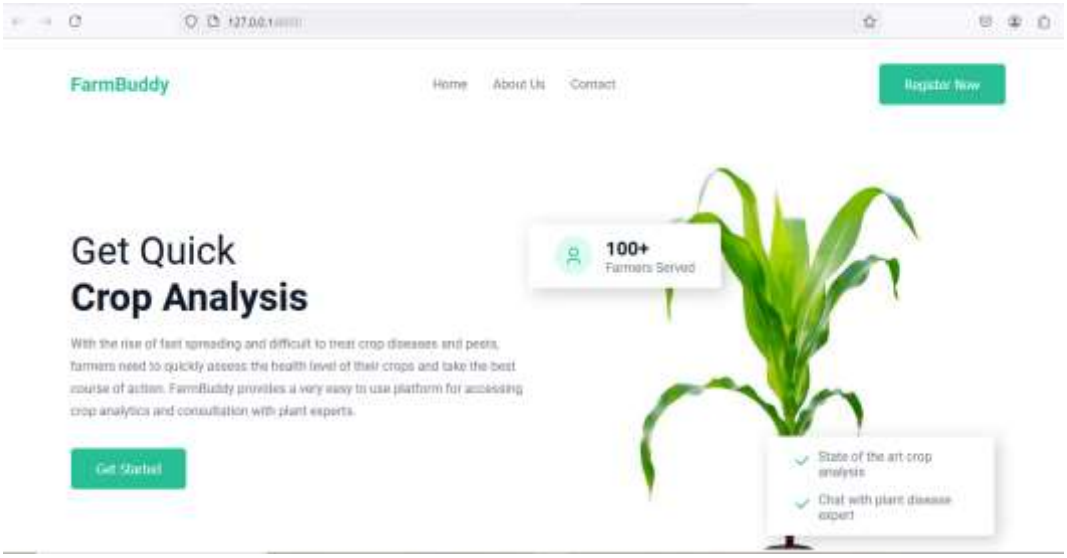


Figure 6.1 User Interface for the Application

The FarmBuddy web application's user interface is shown in Figure 6.1. Aimed to provide a user experience that is both easy to use and interesting, the interface was created with a minimalistic and contemporary style. At its core, the application consisted of a top navigation bar, a header with an alert message, and a few interactive components.

The Register Now button was prominently displayed on the right side of the navigation bar to encourage people to join the platform. The inviting slogan that reads, Get Quick Crop Analysis, emphasises the platform's objective. At the bottom was a paragraph that provided a concise description of how the increasing threat of pests and crop diseases necessitated rapid assessments of crop health.

Users could easily access the dashboard by clicking the Get Started button on the header's left side; this served as a clear call to action for anyone interested in starting to use the application. The platform's agricultural theme was further reinforced by an image of maize crops displayed on the right side of the page. Two content blocks accompanied the image. One block had the figure 100+ Farmers Served, which emphasised the platform's impact, while the other block featured a human icon. To

make the platform's functions simple to understand and use, the design effectively integrated interactive components like buttons and navigation with instructive content.

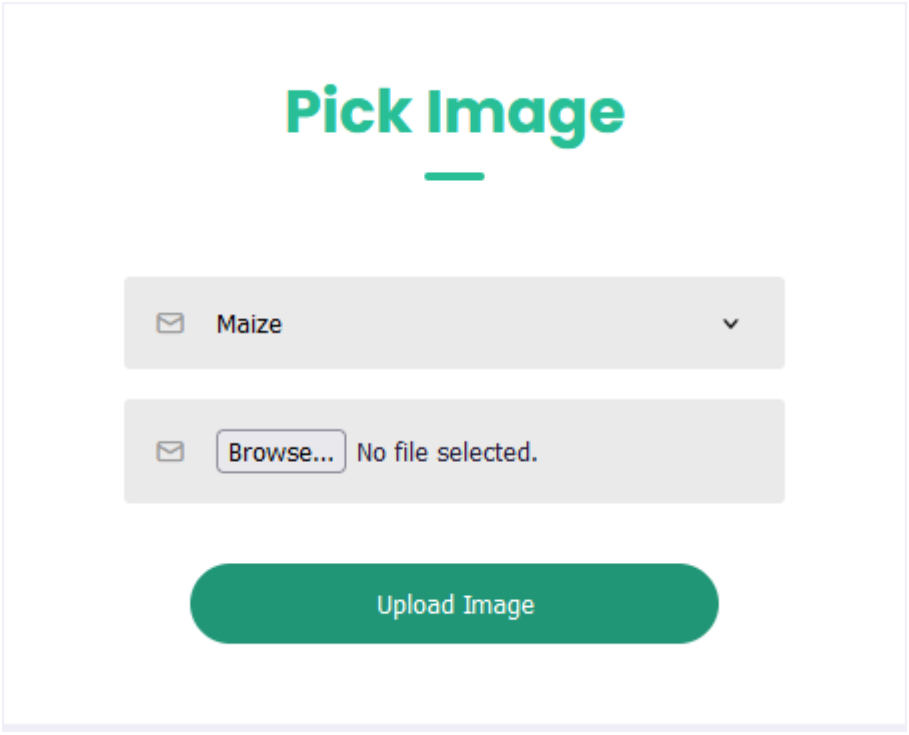


Figure 6.2 Upload Image

The user interface shown in Figure 6.2, which was created using HTML code, had an easy-to-understand layout with the goal of making image analysis easier. A simple form that allowed users to upload an image and select the analysis model associated with the uploaded image.

An eye-catching headline, Pick Image, served as a clear prompt for users to perform the main desired action. Two important input fields were accessible through the form that was located below the header. The first input box had an icon-based dropdown menu with three preloaded models to select from: Maize, Tomato, and Pest. The system was able to choose the model to use for image analysis based on this option.

To facilitate analysis, a second input box allowed users to upload an image file. By using a conventional file input button, the form enabled users to access their local devices in search of an image.

At the bottom of the form, when the user had finished entering the information required, there was a big button that said Upload Image that was used to submit the

data. The layout was placed within a container that consolidated the form information, making it easy to access on different devices. An appealing visual design and icon support were provided by the external stylesheets included in the code, which contributed to an up-to-date and user-friendly interface.

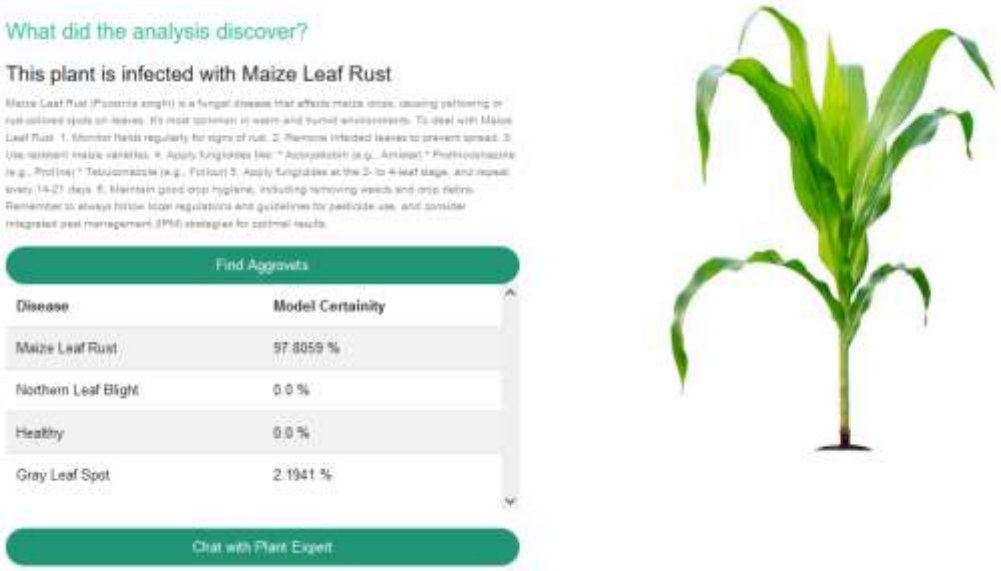


Figure 6.3 Image Analysis for Maize Diseases

Figure 6.3 was a result of clicking the upload image button, as shown in Figure 6.2, after the user provided all the details. Figure 6.3 user interface used a responsive layout, with the left and right halves of the screen separated by a container. You could access several features, including the Dashboard, Analysis, User, Chat, and Aggrovets, using the navigation bar on the left side of the page. The icons and links for each of these elements were clearly displayed. It was clear that the user was now on the Analysis page, as it was highlighted as an active link.

The results of the analysis were shown in the content section on the right side of the interface. What did the analysis discover? was the first question posed in the header. After that, a precise diagnosis was made, indicating that the plant had been affected by Maize Leaf Rust. The symptoms, causes, and treatments for Maize Leaf Rust were described in depth here. Azoxystrobin, Prothioconazole, and Tebuconazole were among the fungicides listed, along with recommendations for how often and when to use them.

A table showing the model's confidence levels for several crop diseases was included under the diagnosis and recommendations. These crop diseases were Maize Leaf Rust, Northern Leaf Blight, Healthy, and Gray Leaf Spot. A confidence level of 97.6456% was recorded for maize leaf rust, a low percentage for northern leaf blight and gray leaf spot, and zero for healthy.

Users located local agricultural vendors by clicking the Find Aggrovets button, and they could communicate with a plant expert by clicking the Chat with Plant Expert button. Another button, Analyse Image, which led to an image upload form, gave users the ability to add additional images for analysis.

The analytical context was graphically reinforced on the right side of the interface with a huge image of a maize plant. The image was the focal point, dominating a large chunk of the right side. Streamlining the process of seeing analytical results, consulting experts, and locating agrovets, this clean, organised, and responsive design made navigation easy, hence ensuring a seamless user experience [268].

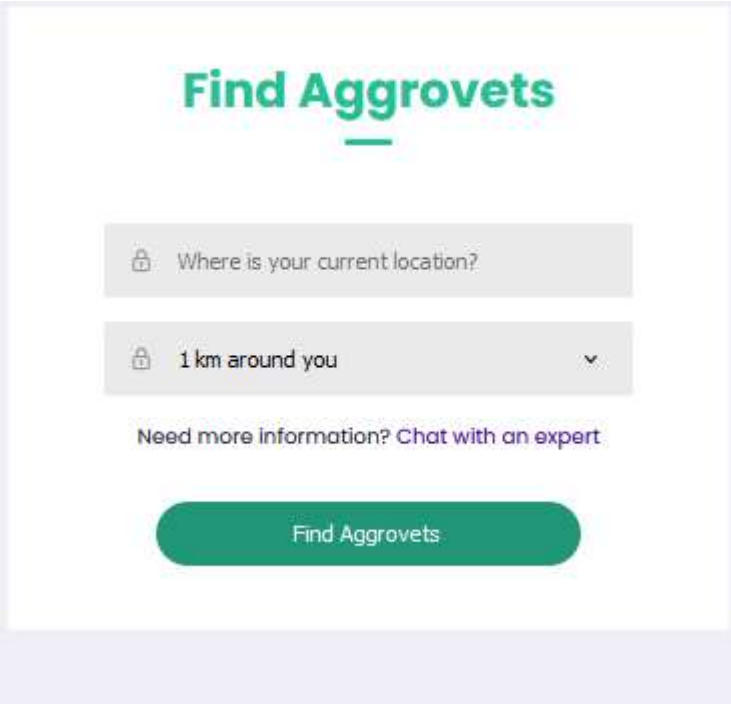


Figure 6.4 Find Agrovets

To make it easier to find neighbouring agrovet businesses, the user interface for Figure 6.4 was built around a centred, basic form within a clean and straightforward

homepage. The form box was the focal point of the design, which kept the emphasis on the offered functionality.

Find Agrovets was the header that showed up at the top of the form, so it was easy to see what the interface was for. Below this header, users could see a form that asked for their present location and a search radius, allowing them to locate agrovets within that distance.

There were two fields for data entry on the form. In the first, a padlock-shaped text input form asked users to fill in their current location with the question, Where is your current location? The next input that allowed users to specify their desired search radius was a dropdown menu that was also denoted with a padlock emblem. The three distances displayed by the dropdown menu—1 km around you, 5 km around you, and 10 km around you—were designed to meet distinct proximity requirements.

A link was provided underneath the input fields, allowing users to Chat with an expert for more information. By connecting users to a chat service, this connection added an extra degree of support, guaranteeing that they could get more help if needed. The Find Agrovets button at the very bottom of the form allowed users to input their search criteria. The button was designed to be highly visible and easily accessible, ensuring that users clearly understood its purpose.

The interface's design prioritized functionality without adding unnecessary components, making it simple and easy to use. With the help of icons and placeholders, the form became more user-friendly, and the chat feature provided additional support. The primary focus of the web application was to facilitate the efficient and rapid location of agrovets by users, taking into account their present position and the search radius they had set.



Figure 6.5 Display of Agrovets Around Thika Town

Using an interactive map, users could locate neighboring agrovets as shown in Figure 6.5's user interface. The layout had a sidebar for navigation and a main text section that prominently displayed the map. A well-balanced navigation menu and map interface contributed to the page's clean and organized design.

The left side of the web application included a navigation bar with the FarmBuddy logo at the top. Below it, there were links to several areas such as the Dashboard, Analysis, User, Chat, and Agrovets. As the active page, the Agrovets link indicated that users were currently viewing content relevant to Agrovets. Each menu item was accompanied by an icon, enhancing visual clarity and ease of navigation.

The primary content section, labelled custom content, was distinguished by its interactive Google Maps feature. Initially zoomed in to a level of 15, the map was centred (latitude: -1.0387569, longitude: 37.0833753) and showed a detailed perspective of the agrovets. The markers on the map indicated the locations of various agrovet stores.

At specific locations across the map, users could find markers that, when clicked, provided the name and contact information of an agrovet. Among the listed agrovets were Mgt Agrovet (0743 898978), Athi Agrovet Suppliers (0751 646486), and Tracy Agrovet (0792 302760). The position of each agrovet was clearly shown on the map, so users could simply determine how close they were and how to get in touch with them.

With limited text and alternatives at the top of the content section, the map could be seen clearly on the right-hand side of the page, making it the primary focus. The page's simplicity and usability were maintained since no other elements were included below the map.

The map layout allowed users to visually explore agrovet locations, while the navigation bar provided easy access to other sections of the platform. The user interface was very helpful since users were able to efficiently locate agrovet stores and access contact information directly from the interactive map, which was enhanced with markers and instructive details [269].



Figure 6.6 Plant Expert

Figure 6.6 illustrates a chabot user interface featuring a modern look and feel, navigation tools, and interactive conversation capabilities. The user interface featured a chat panel on the right side and a navigation sidebar on the left.

In the navigation sidebar at the top, you could see a logo labelled FarmBuddy, and then you could see a toggle button that allowed users to expand or collapse the sidebar. The toggle was accompanied by links to other sections of the application, such as Dashboard, Analysis, User, Chat, and Agrovets. By highlighting the currently active page, Chat, the users could see the current user location within the application. A Close

button with an icon to indicate logging out or leaving the application was located at the base of the sidebar.

A chat interface served as the primary content area, facilitating user interaction. The chat panel's header displayed a rounded avatar of the chatbot, the name ChatBot, and the slogan Ask me anything. Next to the avatar, there was a green online status symbol that showed when the chatbot was available.

The dialogue bubbles displaying previous messages were located in the chat area, which was placed at the centre. The chatbot's messages, including the welcome Hello there, how may I help you? Were shown with an avatar and a timestamp. Text, profile symbols, and timestamps in the discussion bubbles were visually distinct, following standard practice for chat applications.

Located at the very bottom of the chat window was the area for entering messages. The user could write their message into the input area and then click the Send button, which included a location-arrow icon, to submit it. The researcher made sure that the input area was easy to use and that all of the form components were responsive and user-friendly. The chat window also included other buttons like Contact Us and Analyze Image button that offered more functionality beyond just chatting [270].

6.4 Summary

The process of integrating a web-based application that supported stakeholders in crop disease and pest management with a hybrid model of Convolutional Neural Network (CNN) and Support Vector Machine (SVM) was fully explained in this chapter. This chapter provided a detailed account of the model's development, with an emphasis on the use of a Convolutional Neural Network (CNN) for feature extraction from pests and crop disease images, and a Support Vector Machine (SVM) for classification. By combining the advanced pattern recognition capabilities of Convolution Neural Networks with the efficient classification technique of Support Vector Machine, this hybrid approach improved the accuracy of crop disease and pest classification.

Stakeholders were able to engage with the web application's responsive and user-friendly interface, which was developed using HTML, JavaScript, and CSS for the front end. Using FastAPI as the backend enabled the application to perform real-time image analysis and support batch processing for users who wished to upload multiple

images simultaneously. Affected crop images and pest images were uploaded by stakeholders using the interface, and the model automatically classified the images using the extracted features. The model provided the classification results and the accuracy of the model for each image. The accuracy of the classification could be seen clearly and transparently by the users.

The model went a step further by offering personalized recommendations for crop disease and pest control after identifying the image. Based on the classification, the system recommended fungicides and pesticides for crop diseases and pests. Stakeholders, particularly farmers, were able to use this functionality to better understand and control crop diseases and pests. To lessen the likelihood of crop losses and increase agricultural output generally, the recommendations were important for supporting prompt measures.

The web application integrated Google Maps to support stakeholders in locating local agrovet stores, in addition to its classification and recommendation functions. The application leveraged location-based services, enabling users to find the nearest agrovet based on their current location. In addition to pinpointing the exact position of the agrovet on the map, this function also included useful information like addresses and phone numbers, facilitating the acquisition of pesticides and fungicides.

A full-fledged decision-support system for crop diseases and pest management was developed by integrating the web application with the hybrid CNN-SVM model. Users were able to improve their crop management skills with the help of the application's data-driven decisions made possible by a hybrid deep learning model, location-based services, actionable recommendations, and real-time data processing. By providing an easy and effective way to deal with issues related to pest and crop disease control, this web application marked a significant advancement in agricultural technology.

CHAPTER SEVEN
VALIDATION OF MODEL ROBUSTNESS AND GENERALIZABILITY

7.1 Introduction

The validation of the model, with its wide variety of crop diseases and pests, was important to assess the robustness and generalizability of a deep learning model to guarantee its efficacy in real-world applications. This chapter focused on the main objective of testing the developed hybrid CNN and SVM model's robustness and generalizability across different types of crop diseases and pest variations that farmers faced in the real-world scenario. The purpose of the validation method was to assess the model's robustness in real-world agricultural scenarios, including datasets containing unseen variations of crop diseases and pests.

Variability in pest and crop disease symptoms was a common result of the complexity of agricultural ecosystems, which included variances in crop species, development phases, environmental conditions, and geographic locations, among other things. It was important to make sure the hybrid CNN-SVM model was still accurate and robust under these conditions. The model's predicted accuracy and generalizability across multiple settings were evaluated during validation. This was because crop leaf diseases had similar symptoms, while pests had drastically different appearances.

This was accomplished by testing the model on a large dataset that extended beyond the confined training environment. To ensure robustness, the researcher used images depicting crops at multiple stages of disease and pests, taken under diverse lighting, weather, and soil conditions. The validation process assessed the model’s capacity to accurately differentiate among various crop disease and pest classes. To further evaluate the model's usability and performance in a real-world environment, the researcher also looked at its interaction with agricultural stakeholders, such as extension officers and farmers, who uploaded crop images through the web application.

The validation aimed to prove that the hybrid CNN-SVM model could withstand the inherent unpredictability of real-world agricultural conditions while still producing reliable predictions and high accuracy. This chapter provided a thorough analysis of

the model's capabilities, shortcomings, and room for improvement based on the results and insights from the validation [271].

7.2 Validation Methodology

The validation technique focused on testing the developed hybrid of CNN and SVM models' robustness and generalizability over a wide range of crop diseases and pest variations. Using real-world and secondary data for different crop diseases and pests, the researcher aimed to assess how well the model performed. This step was important in ensuring the model performed well on test datasets other than the training dataset.

The first step was to obtain a large number of test images depicting different crop diseases and pests from diverse agricultural contexts. To ensure the model could handle real-world scenarios, such as varying light conditions, different stages of crop health, and diverse pest life cycles, this diverse dataset was important. To further test the model's generalizability, the images were selected to include rare and less common crop diseases and pests.

A k-fold cross-validation method was used for validation, which included dividing the dataset into k separate subsets. To guarantee that every subset was used for testing at least once, the procedure was repeated k times. The model was trained on k-1 subsets and then tested on the remaining subset. To reduce the possibility of overfitting and guarantee good generalizability to new data, this approach offered a thorough assessment of the model's performance across several data subsets. In addition, the model's reliability and accuracy in real-world scenarios were evaluated by comparing its predictions to data that had been labelled by experts.

To systematically evaluate the model's performance, statistical measures including recall, precision, F1 score, and accuracy were computed. These measures provided insight into the model's robustness across different datasets, its sensitivity to false positives, and its ability to accurately identify crop leaf diseases and pests. The results showed that the model, a combination of Convolutional Neural Network (CNN) and Support Vector Machine (SVM), was accurate and generalizable enough to be used in real-world agricultural scenarios [272].

7.3 Generalizability Across Diverse Crop Diseases and Pest Variations

The chapter presents the results of the study that validated the robustness and generalizability of the proposed hybrid CNN and SVM model across diverse crop disease types and pest variations. To ensure the model's applicability outside the controlled dataset used during training, this validation focused on examining its capacity to reliably identify diverse crop diseases and pest varieties in real-world agricultural scenarios.

To validate the model, the researcher used data from a wide range of agricultural settings, including varying latitudes, temperatures, and crop types. Various crop diseases and pests, including those that affect maize and tomato, were included in this dataset. The model's performance was measured based on its accuracy in classifying images of crop diseases and pests. This allowed the model to generalise over a wide range of scenarios, rather than being limited to specific crop disease varieties or geographic regions.

Testing the hybrid model's flexibility to unforeseen scenarios also confirmed its robustness. To achieve this, the researcher augmented the model's training dataset with images of crop diseases and pests. The model's ability to accurately identify and classify these new instances validated its practical applicability. Even when presented with variations in image quality, lighting conditions, and partial occlusions, the model maintained a high level of accuracy.

The hybrid CNN and SVM model was validated for its ability to enhance decision-making in crop leaf diseases and pest management through rigorous testing, which demonstrated its robustness and effectiveness in real-world settings. The model's ability to manage various crop diseases and pests and provide reliable recommendations for pest and crop disease control showed its relevance to stakeholders in crop leaf diseases and pest management, as confirmed by the results [273].

7.4 Real World Agricultural Setting Validation

This section addresses in detail the specific objective of testing the hybrid CNN and SVM model's robustness and generalizability across diverse crop disease types and

pest variations in real-world agricultural settings. The researcher validated the model to ensure it could confidently handle real-world agricultural scenarios beyond the controlled training environment.

To achieve this, the model was evaluated using a range of datasets obtained from real-world agricultural settings. Images of crop diseases, including maize and tomato, showing a wide range of disease signs and pest infestations, were included in this dataset. To reflect the unpredictable and uncontrollable nature of real-world agricultural settings, the images captured represented diverse conditions, including variations in lighting, angles, and image quality.

The ability of the model to maintain its high classification accuracy when subjected to these diverse inputs was used to evaluate its robustness. Weevils, locusts, aphids, and rust were among the pests against which it was tested. Maintaining a high degree of accuracy, the hybrid CNN and SVM model consistently performed well in identifying and classifying various pests and crop diseases. Its generalizability was further validated by incorporating unseen data of crop diseases and pests that were not included in the training dataset. The model's accurate image classification after including these additional parameters demonstrated its generalizability to a wide range of crop diseases and pests.

To further ensure the quality and effectiveness of the model's outputs, the researcher cross-checked them with an expert in crop diseases and pest management. This included the model's treatment recommendations, which included pesticides and fungicides. Integrating the web application, which allowed stakeholders in real-world scenarios to upload images of crop diseases and pests, was also part of the support process. Agronomists and agricultural experts confirmed that the classifications and recommendations generated by the model's accurate processing of the crop diseases and pest images were correct.

The model's robustness and generalizability were shown by real-world testing, which proved that it could be implemented consistently in diverse agricultural conditions. This helped farmers and agricultural stakeholders to efficiently manage crop diseases and pests. The validation process confirmed that the model worked as intended in real real-world environment, making it applicable to crop diseases and pest management.

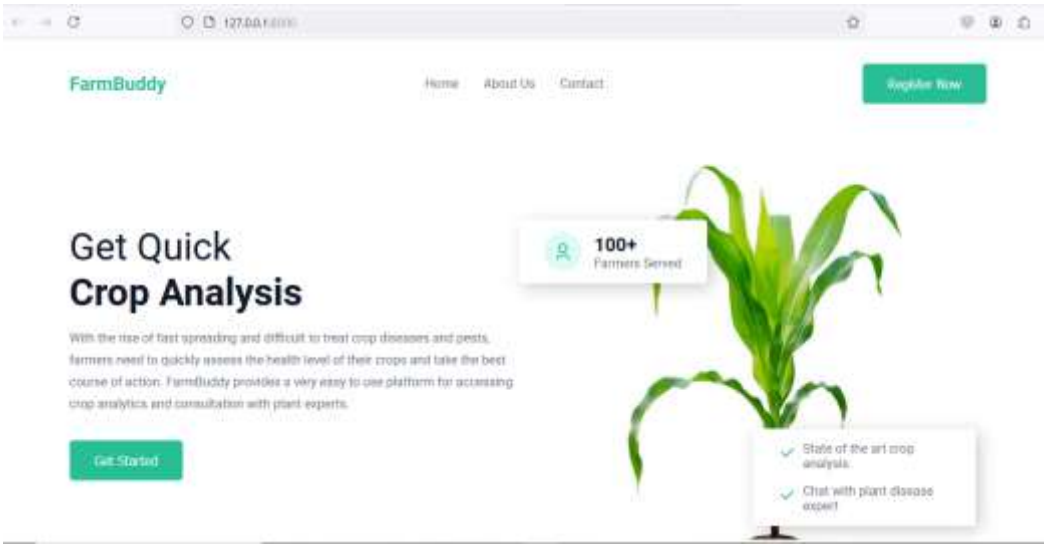


Figure 7.1 User Interface for the Application

Figure 7.1 shows the user interface of the FarmBuddy web application. The interface's simple and modern look was an attempt to give an engaging and easy-to-use user experience. The main features of the application included a top navigation bar, a promotional message header, and other interactive components.

On the right side of the navigation bar, there was the Register Now button, which was used to sign up users on the platform. The platform's goal was highlighted by a large, appealing phrase that occupied a significant section of the top area of the interface and read, Get Quick Crop Analysis.

There was a prominent call to action for anybody interested in beginning to use the application, and users could simply enter the dashboard by clicking the Get Started button on the left side of the header. The image of the maize crop on the right side of the page further reinforced the platform's agricultural theme. Along with the image were two blocks of information. A human symbol appeared on one block, and the number 100+ Farmers Served on the other, both of which served to emphasise the impact of the platform. The design integrated useful content with functional elements like buttons and navigation to make the platform's features easy to understand and use.

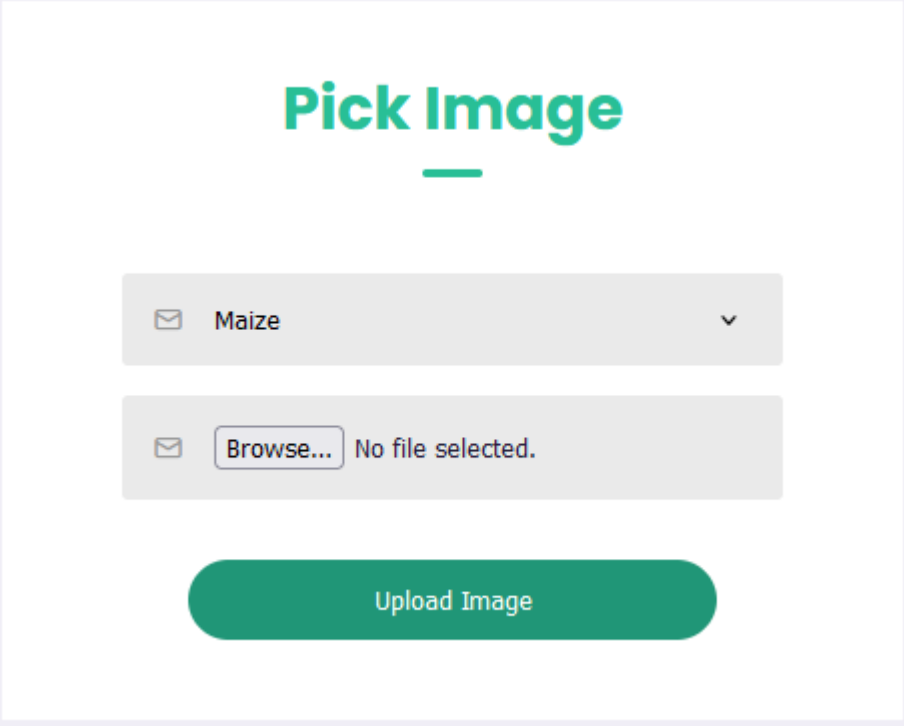


Figure 7.2 Upload Image

Figure 7.2 shows the result of an HTML-coded user interface that aimed to simplify image analysis via the use of an intuitive layout. As part of its user-friendly design, the interface included a simple form where users could select an image and its corresponding analysis model.

An attention-grabbing title, Pick Image, clearly indicated the primary action that was meant to be taken by users. Underneath the header, there was a form with two important input areas. There were three preset models, Maize, Tomato, and Pest, accessible through an icon-based dropdown menu in the first input field. This setting allowed the system to select the appropriate model for image analysis. Users were able to attach image files to support the analysis process using a second input box. The form allowed users to access their local devices to search for images by using a standard file input button.

The user uploaded the image by clicking the Upload Image button at the bottom of the form once they had completed entering the two pieces of information. Its responsive design made it easier to include the layout in a container that included all of the form data, making it accessible from any device. A more modern and user-friendly interface

was made possible by the external stylesheets that were included in the code. These stylesheets provided an attractive visual design and support for icons [274].

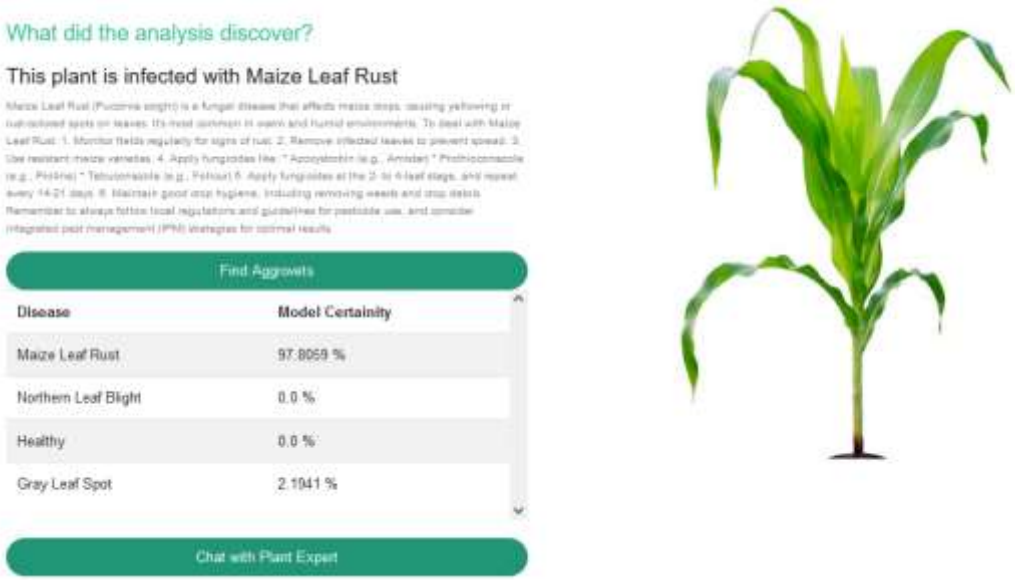


Figure 7.3 Image Analysis for Maize Diseases

Figure 7.3 was a result of clicking the upload image button, as shown in Figure 7.2, after the user provided all the details. Figure 7.3 user interface used a responsive layout, with the left and right halves of the screen separated by a container. You could access features, including the Dashboard, Analysis, User, Chat, and Aggrovets, using the navigation bar on the left side of the page. The icons and links for each of these elements were clearly displayed. It was clear that the user was now on the Analysis page, as it was highlighted as an active link.

The results of the analysis were shown in the content section on the right side of the interface. What did the analysis discover? was the first question posed in the header. After that, a precise diagnosis was made, indicating that the plant had been affected by Maize Leaf Rust. The symptoms, causes, and treatments for Maize Leaf Rust were described in depth here. Azoxystrobin, Prothioconazole, and Tebuconazole were among the fungicides listed, along with recommendations for how often and when to use them.

A table displaying the model's confidence levels for various crop diseases was included under the diagnosis and recommendations section. These conditions were Maize Leaf Rust, Northern Leaf Blight, Healthy, and Gray Leaf Spot. A confidence

level of 97.6456% was recorded for maize leaf rust, 2.1941% for gray leaf spot, 0.0% for northern leaf blight and healthy.

Users could locate local agricultural vendors by clicking the Find Aggrovets button, and they could converse with a plant specialist by clicking Chat with Plant Expert button. Another button, Analyze Image, which led to an image upload form, gave users the ability to add additional images for analysis.

The analytical context was visually reinforced on the right side of the interface with a large image of a maize plant. The image was the focal point, dominating a large chunk of the right side. Streamlining the process of seeing analytical results, consulting experts, and locating agrovets, this clean, organised, and responsive design made navigation seamless [275].

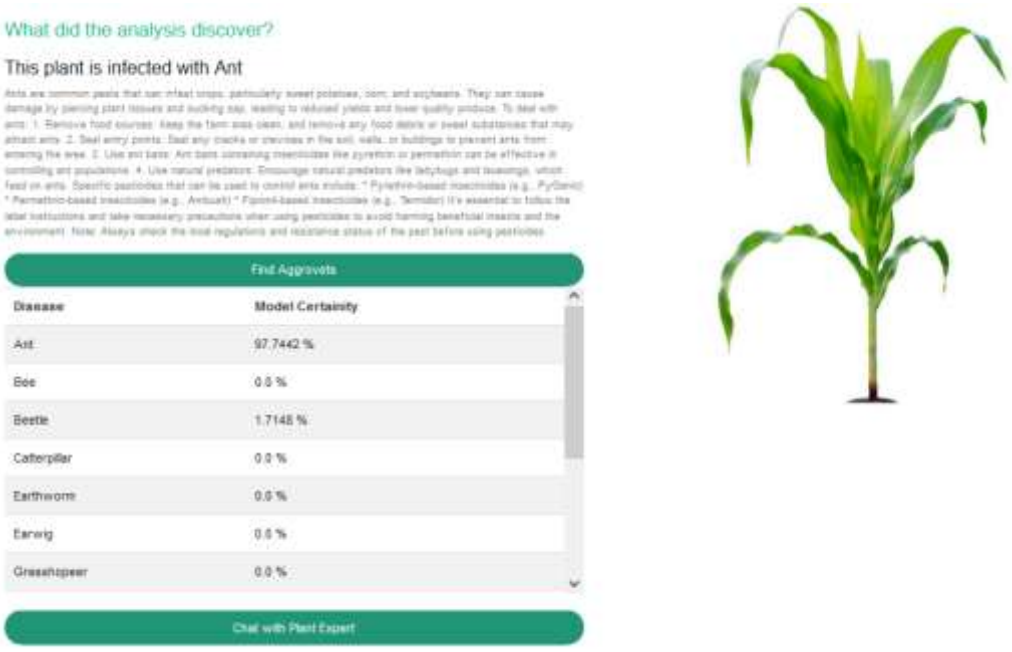


Figure 7.4 Image Analysis for Pest

Figure 7.4 user interface, built using HTML code, featured well-structured components arranged in a clean and straightforward layout for crop diseases and pest analysis. The application's user interface included a left-hand navigation bar and a two-column main content area.

Embedded inside a sidebar on the left side was a navigation menu branded with the FarmBuddy logo and title. The areas that were accessed using this menu included the Dashboard, Analysis, User, Chat, and Agrovets. By using a separate visual style to emphasize the selected page, Analysis, the users were able to clearly differentiate it from other available options.

The Analysis page's main content area was located on the right side of the interface. The analyzed pest image was found to be an ant, as stated in the subsection that followed the top heading What did the analysis discover?. The next paragraph provided detailed information on the ants' role as pests and the damage they do to crops. Included in this was guidance on dealing with the pest in a variety of ways, including ways to avoid it, like eliminating food sources, and ways to deal with it, such as sealing entry points, using ant baits, and encouraging natural predators. Additional examples of pesticides that may be used to reduce the ant population were included in the text, along with cautions regarding the environmental impact and usage restrictions.

The model's analysis of various pests was summarized in a table displayed below. The table included columns for the pests that were identified and the confidence levels associated with the model’s predictions. Model certainty for the identification of the Ant pest was 97.6757%; however, for the identification of the bee and grasshopper pests, there was 0% confidence. The detailed summary of the findings was presented in this table, enabling the users to quickly understand the analysis.

The interface's functionality and user engagement were enhanced with the inclusion of a sequence of buttons. Users could analyse images with the click of one button and locate nearby agrovet providers with the swipe of another. Further help was provided by users by initiating a chat with an expert using the Chat with Plant Expert button. The design became more user-friendly with the addition of these interactive components, which allowed users to readily act upon the information given.

In general, the interface was ease of use and interactive, with a focus on improving decision-making for crop diseases and pest management through intuitive navigation, clear data presentation, and user-friendly interactions [276].

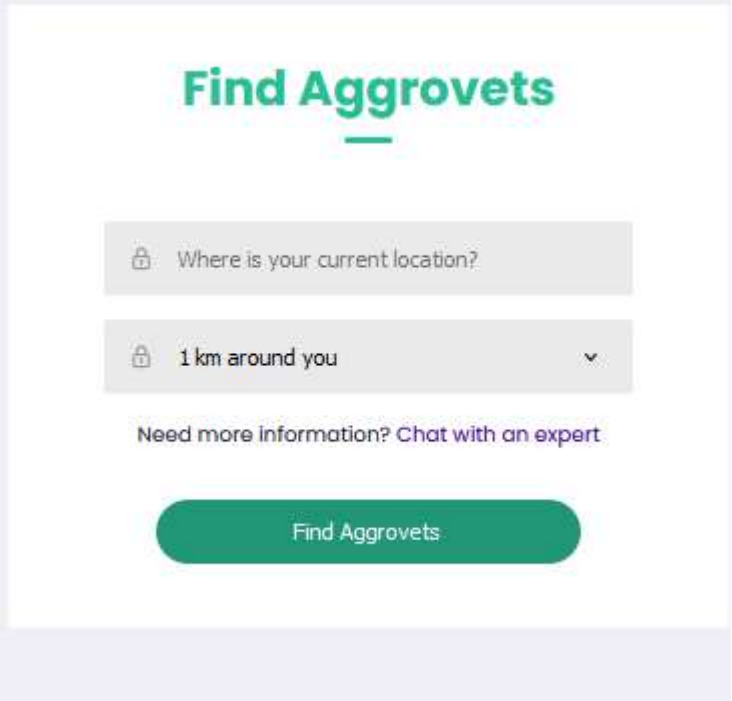


Figure 7.5 Find Agrovets

A centred, simple form inside a clean and clear user interface was used to design the user experience for Figure 7.5, making it easier to identify nearby agrovet shops. The design maintained the form box as its focus point, effectively highlighting its intended functionality.

The presence of the 'Find Aggrovets' heading at the top of the form made the interface's purpose instantly recognisable to users. By filling out the form located beneath the heading that prompted users to enter their current location, users were able to search for agrovets within a defined radius.

In the first field, users were prompted to enter their current location through a text input styled with a padlock icon and the question, 'Where are you right now?' A second input, also represented by a padlock icon, was a dropdown menu that allowed users to select the desired search radius. In order to accommodate different needs for proximity, the drop-down menu offered three different lengths: 1 km around you, 5 km around you, and 10 km around you.

The input fields were accompanied by a link labelled Chat with an Expert, allowing users to seek additional information. This link provided an additional layer of support

by linking users to a chat service, ensuring that they could get more support if they needed it.

At the very bottom of the form, a large and conspicuous Find Aggrovets button allowed users to input their search parameters. For customers to understand its function, the button was designed to be visible and easily accessible.

Simplicity and ease of use were key goals in the interface's design process, which emphasised functionality over unnecessary elements. Icons and placeholders made the form easier to use, while the chat with expert function added more support. With the current location and the search radius the users had specified, the major goal of the application was to help users quickly and easily find agrovets.



Figure 7.6 Display of Agrovets Around Thika Town

Users could find nearby agrovets using the interactive map interface displayed in Figure 7.6. The design had a navigational sidebar and main content area that displayed the map at the forefront of the interface. The page's clean and organized design was supported by a well-balanced navigation menu and map interface.

There was a navigation bar on the left side of the web application that included the FarmBuddy logo at the top. It was followed by links to several sections, including the Dashboard, Analysis, User, Chat, and Agrovets. The fact that the Agrovets link was the active page meant that people were looking at Agrovets-related stuff at the moment. It was simple to see what you were doing since icons were placed next to the menu items.

The main content area, labelled custom content, stood out due to its interactive Google Maps functionality. At first, with the zoom level set to 15, the map displayed a detailed view of the regions surrounding a specific point (latitude: -1.0387569, longitude: 37.0833753). The map displayed markers indicating the specific locations of various agrovet shops.

Users could locate agrovets' names and contact information by clicking on markers placed in specific locations across the map. Among the listed agrovets were MGT Agrovet (0743 898978), Athi Agrovet Suppliers (0751 646486), and Tracy Agrovet (0792 302760). Users could easily view the location of each agrovet on the map and get in contact with them based on the provided information.

With minimal content and elements at the top of the page, the map became the focal point on the right side of the interface. The exclusion of additional content below the map helped preserve the page's clean, minimalist layout and user-friendly experience.

Easy access to different sections was provided through the navigation bar, and the map layout enabled users to visually explore agrovet locations. The interactive map, enhanced with instructive markers, allowed users to efficiently identify agrovet shops and retrieve contact information directly, significantly enhancing the overall user experience.



Figure 7.7 Plant Expert

A chatbot with a modern design, navigational tools, and interactive conversational features is shown in Figure 7.7's user interface. On one side of the user interface was a chat panel, while on the other was a navigation sidebar.

There was a FarmBuddy logo at the top of the navigation sidebar, along with a toggle button that allowed users to open or close the sidebar. Upon toggling, the sidebar displayed navigation links to various sections of the platform, such as Dashboard, Analysis, User, Chat, and Aggrovets. To help users understand their current position within the interface, the active page, such as Chat, was visually highlighted. At the very bottom of the sidebar, there was a close button accompanied by an icon representing logging out or exiting the application.

The upper section of the chat panel featured the chatbot's circular avatar image, its name ChatBot, and the slogan Ask me anything!. A green status icon beside the avatar indicated that the chatbot was online and available.

In the middle of the chat area, there were the discussion bubbles that showed past messages. Hello there, how may I help you? It was only one example of the chatbot's greetings shown beside a timestamp and avatar. In keeping with best practices for chat applications, the conversation bubbles' text, profile icons, and timestamps were visually different.

The users could find the space to type in their messages at the bottom of the chat window. After typing their message into the input field, users could submit it by clicking the Send button, which had a location-arrow symbol. All of the form's components, including the input area, were tested for responsiveness and user-friendliness by the researcher. There were other options in the chat window that allowed users to do other tasks, such as Contact Us and Analyze Image [276].

7.5 Comparative Analysis with Existing Models

The comparative study aimed to evaluate the robustness and generalizability of the developed hybrid model by comparing it to existing models, including DenseNet, AlexNet, EfficientNet, VGG, and CNN. The developed hybrid model integrated a Convolution Neural Network (CNN) with a Support Vector Machine (SVM). To evaluate how well these models worked, the researcher utilised a large dataset that included a wide range of crop diseases and pest variations seen in real-world farming.

Images depicting various crop diseases and pests were part of the extensive dataset used in the research. To test the models' generalizability, this dataset was important. Accuracy, precision, recall, F1 score, and confusion matrix were among the performance metrics that were carefully evaluated throughout each model's testing and training phases. To make sure that all of the models could deal with changes in image quality, data augmentation methods were used to make the models more robust.

After careful analysis, it was found that DenseNet's exceptional performance could be attributed to its unique architecture, which enabled better gradient flow and effective use of parameters. The model's accuracy was especially impressive when handling crop diseases and pests. However, AlexNet's feature extraction capabilities were limited by its relatively simplistic architecture and fewer convolutional layers, which made it a poor generalizable model when subjected to unseen data.

Another top-performing model was EfficientNet, a well-known scaling approach that found the optimal balance between model size and performance. Its efficiency allowed it to outperform other models, providing comparable accuracy at a lower computational cost. Despite its reputation for depth, VGG had an overfitting problem, especially on smaller datasets, which made it less reliable in practical settings. Although it served as a foundation, the traditional CNN model couldn't keep up with the new architectures and exhibited inconsistent performance that required extensive tuning and data preprocessing.

According to the comparison results, the hybrid CNN and SVM model outperformed the other models across all performance indicators. The improved feature extraction and classification made possible by its design led to increased accuracy while handling a wide range of pests and crop diseases. It is worth mentioning that the hybrid model successfully adapted to new, unknown data without experiencing major performance declines, maintaining its robustness and generalizability.

Validating the robustness and generalizability of models in crop diseases and pest management was crucial. In addition to surpassing more conventional models such as DenseNet, AlexNet, EfficientNet, VGG, and CNN, the developed model that combined CNN and SVM showed remarkable adaptability to the intricacies of real-

world agricultural problems. Stakeholders in crop disease and pest management can build on the results of this study to improve decision support systems [277].

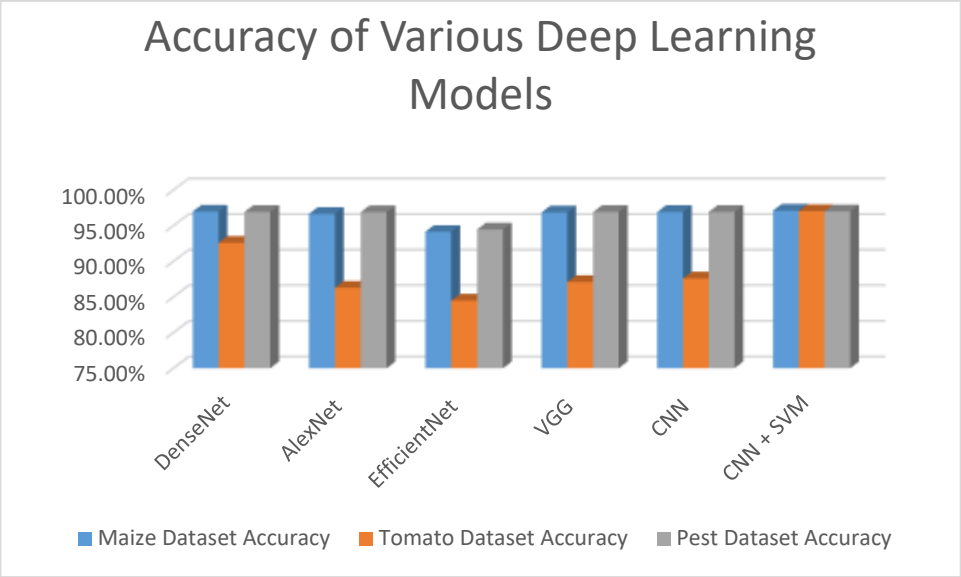


Figure 7.8 Accuracy of Various Deep Learning Models

With accuracy values reflecting each model's success in classifying the images, Figure 7.8 presents a comprehensive comparison study of the different models' performance on the maize, tomato, and pest datasets. Using the conventional formula, the accuracy was calculated as;

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \times 100 \qquad \text{(Eq. 7.1)}$$

On the maize dataset, DenseNet achieved an accuracy of 96.99%, while on the pest dataset, it was 96.94%. Because of its deep architecture and dense connections, DenseNet was able to efficiently reuse and extract features, leading to its high accuracy. On the other hand, DenseNet's architecture was able to extract highly detailed features from tomato images, which explains why its accuracy on this dataset was at 92.60%.

While AlexNet performed well on the maize dataset (96.66% accuracy) and the pest dataset (96.91%), it showed a significant decline in accuracy on the tomato dataset (86.30%). This drop was attributable to the fact that AlexNet's less complex architecture struggles to differentiate between the tomato dataset's finer details, despite

its success with large-scale data sets like maize and pest. The same reasoning was used to derive the formula used to evaluate its accuracy across different datasets.

$$Accuracy = \frac{Correct\ Maize\ Predictions}{Total\ Maize\ Images} \times 100 \quad \text{(Eq. 7.2)}$$

With accuracies of 94.14% on the maize dataset, 84.50% on the tomato dataset, and 94.47% on the pest dataset, EfficientNet, a network renowned for balancing accuracy and processing efficiency, achieved the lowest scores across all datasets. EfficientNet's poor performance was likely caused by its inability to adequately capture the datasets' complexity, even if its architecture was optimised for computational efficiency. More comprehensive feature extraction may have been required, which EfficientNet's architecture does not fully support, contributing to the reduced accuracy on the tomato dataset.

VGG showed promising results with 96.88% accuracy on the maize dataset and 96.93% accuracy on the pest dataset, but it only managed 87.10% accuracy on the tomato dataset. VGG achieved impressive results, due to its deep architecture, especially on simpler image datasets like maize and pest. On the other hand, the subtleties of the tomato dataset may have been difficult to capture because of VGG's rigid architecture and uniform layer designs.

On the pest dataset, the CNN model achieved 96.94% accuracy, on the tomato dataset it was 87.60%, and on the maize dataset it was 96.93%. The effectiveness of CNN across datasets was due to its capacity to capture hierarchical spatial features. But it had problems with the tomato dataset, much like previous models. This was probably because the tomato images exhibited numerous variations that required more complex feature detection.

No model could match the results of the hybrid CNN + SVM model, which combined the deep learning based feature extraction of CNN with the SVM's ability to handle high-dimensional feature spaces. The accuracy it achieved on the tomato (97.04%), maize (97.13%), and pest (97.02%) datasets was the highest among all models. To determine the model's accuracy, the same method was used as before; however, this

time the results were better, demonstrating that the model could handle more complex and diverse data distributions, especially in the tomato dataset.

$$Accuracy = \frac{Correct\,Tomato\,Predictions}{Total\,Tomato\,Images} \times 100 \quad \text{(Eq. 7.3)}$$

This increased accuracy was a direct result of the improved model's ability to distinguish between similar classes, made possible by combining CNN with SVM.

7.5.1 Confusion Matrix

Beyond overall accuracy, the confusion matrix offered extensive insights into the performance of each classification model, making it an essential tool for evaluation. The performance of a classification model can be described using a confusion matrix, a table that summarises the expected versus actual results across multiple classes. Its four main components were True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Accuracy, recall, and F1-score, among other performance measures, relied heavily on these components.

The confusion matrices of all the models illustrated how effectively they distinguished among the pest, maize, and tomato datasets, in terms of both correctly and incorrectly classified images. The models, such as DenseNet with 96.99% accuracy for maize and AlexNet with 86.30% for tomato, showed the percentage of correct classifications relative to the total predictions for each dataset. This total accuracy was determined by using the following formula;

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \times 100 \quad \text{(Eq. 7.4)}$$

Each model's confusion matrix also revealed its weaknesses, particularly in areas of misclassification. By showing a higher number of misclassified images, the confusion matrix highlighted the increased prevalence of false positives and false negatives, as shown by EfficientNet’s considerably lower accuracy of 84.50% on the tomato dataset. Given its impressive accuracy of 97.13% for maize and 93.76% for tomato in classifying all types of data, the hybrid of CNN and SVM model had a confusion matrix with few false positives and false negatives.

In datasets with greater variability, such as the tomato dataset, analysing confusion matrices made it possible to understand the models' accuracy and identify areas for improvement. By going beyond simple accuracy metrics, the researcher's approach provided a more comprehensive view of each model's robustness.

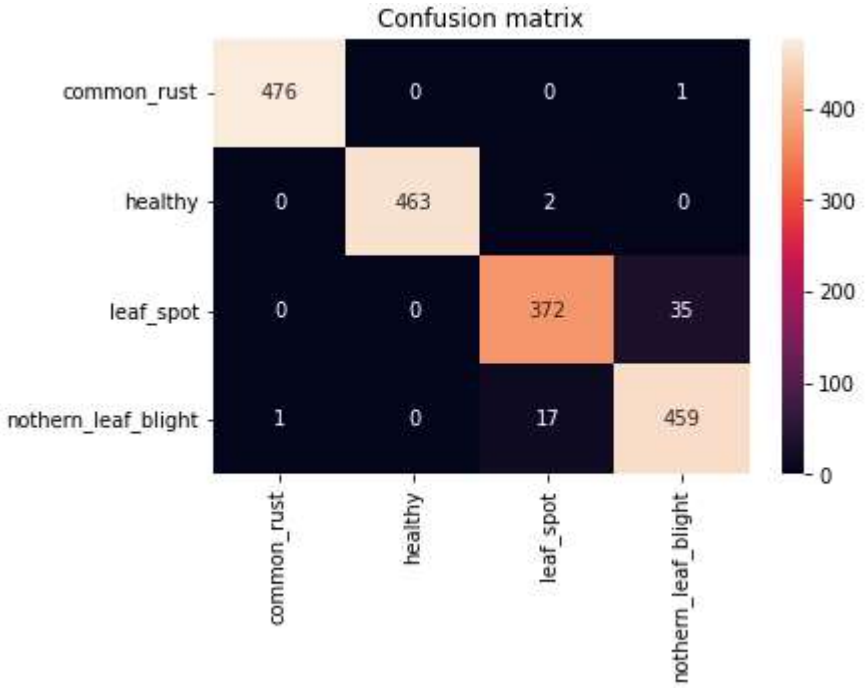


Figure 7.9 Confusion Matrix for DenseNet on Maize Dataset

The analysis of DenseNet's confusion matrix on the maize dataset provided insights into the model's ability to classify four distinct maize leaf diseases: common rust, healthy leaves, leaf spot, and northern leaf blight. The True Positives (TP) were indicated by the diagonal values of the confusion matrix. DenseNet correctly classified 476 instances of common rust, 463 of healthy leaves, 372 of leaf spot and 459 of northern leaf blight.

The model's accuracy, a crucial criterion for validating the model's robustness and generalizability, was computed using these true positive values. DenseNet's overall accuracy on the maize dataset was determined using the following formula;

$$\text{Accuracy} = \frac{\text{TP}_{\text{total}}}{\text{Total Instances}} \times 100 \qquad \text{(Eq. 7.5)}$$

In this case,

$$\text{Total Instances} = \text{Sum of All Predictions} \quad \text{and}$$

$$TP_{total} = 476 + 463 + 372 + 459 = 1{,}770 \left(\text{sum of correct predictions}\right)$$

For example, the accuracy was as follows when the total number of maize images for these four classes was 1,826

$$\text{Accuracy} = \frac{1770}{1826} \approx 96.933\%$$

This remarkable accuracy demonstrated that DenseNet successfully identified the majority of maize leaf disease images. Having said that, the confusion matrix showed some possible shortcomings in the classification of more challenging diseases, such as leaf spot, which had the lowest true positive count of 372. This indicated that DenseNet's ability to generalise well across this particular class was limited due to a higher number of False Negatives (FN) and False Positives (FP) for leaf spot compared to other diseases.

The DenseNet model's strengths and weaknesses were better understood with the help of the confusion matrix. A good example of this was the model’s robustness in identifying healthy leaves (463 true positives) and common rust (476 true positives). But DenseNet had difficulty in classifying leaf spot disease, as demonstrated with the lower true positives for leaf spot, possibly because the image collection was influenced by environmental factors and features that overlapped with other diseases [278].

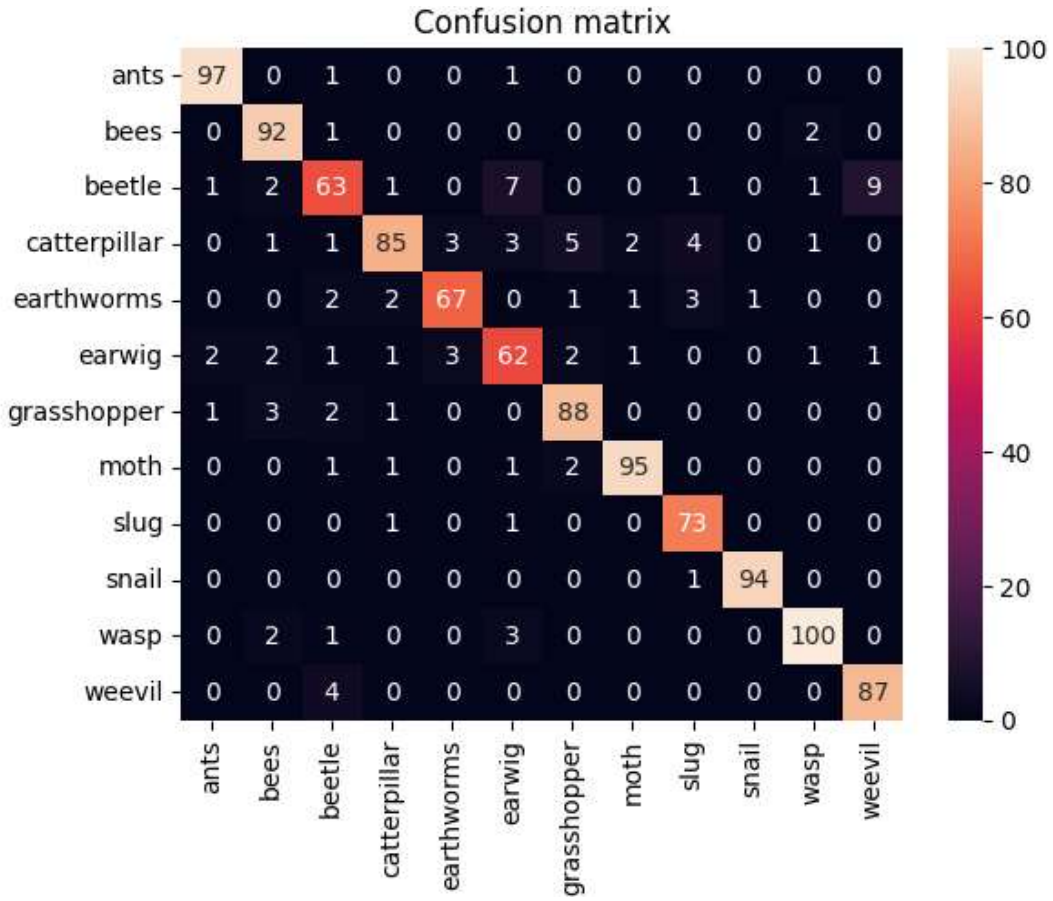


Figure 7.10 Confusion Matrix for DenseNet on Pest Dataset

An important step in confirming the robustness and generalizability of the model was the confusion matrix for DenseNet on the pest dataset, which presented important insights into the model's performance in classifying different types of pests. The number of True Positives (TP) for each pest class was represented by the diagonal values of the confusion matrix. For the pest categories ant, bee, beetle, caterpillar, earthworm, earwig, moth, slug, snail, wasp, and weevil, the corresponding true positive values were 97, 92, 63, 85, 67,62, 95, 73, 94, 100, and 87, respectively. Figure 7.10 provides a direct indication of DenseNet's accuracy for each class by showing how many occurrences of each pest type were accurately classified.

To determine DenseNet's overall accuracy on the pest dataset, the researcher used the following formula;

$$\text{Accuracy} = \frac{\text{TP}_{\text{total}}}{\text{Total Instances}} \times 100 \qquad \text{(Eq. 7.6)}$$

The researcher added up all the right predictions, which were 97 + 92 + 63 + 85 + 67 + 62 + 88 + 95 + 73 + 94 + 100 + 87 = 1,003, then the researcher calculated the total instances by adding up all the predictions.

For this example, there was a total of 1,099 pest images in the dataset (a value that represented all accurate and wrong classifications), and then the accuracy of the model was computed as;

$$\text{Accuracy} = \frac{1003}{1099} \times 100 \approx 91.2645\%$$

The high overall accuracy demonstrated that DenseNet performed very well when it came to classifying the majority of pest classes. However, the confusion matrix revealed that DenseNet struggled to differentiate between certain pest groups, with some categories, including beetles (63 correct predictions) and earwigs (62 correct predictions), showing fewer true positives. This happened because several pests share visual features, which caused a greater likelihood of False Positives (FP) and False Negatives (FN) in such groups.

In contrast, DenseNet performed well with few misclassifications when it came to classifying wasps (100 true positives), ants (97 true positives), and moths (95 true positives). The confusion matrix showed that DenseNet performed well on certain classes but struggled with others. It struggled to distinguish various pest groups that looked visually similar.

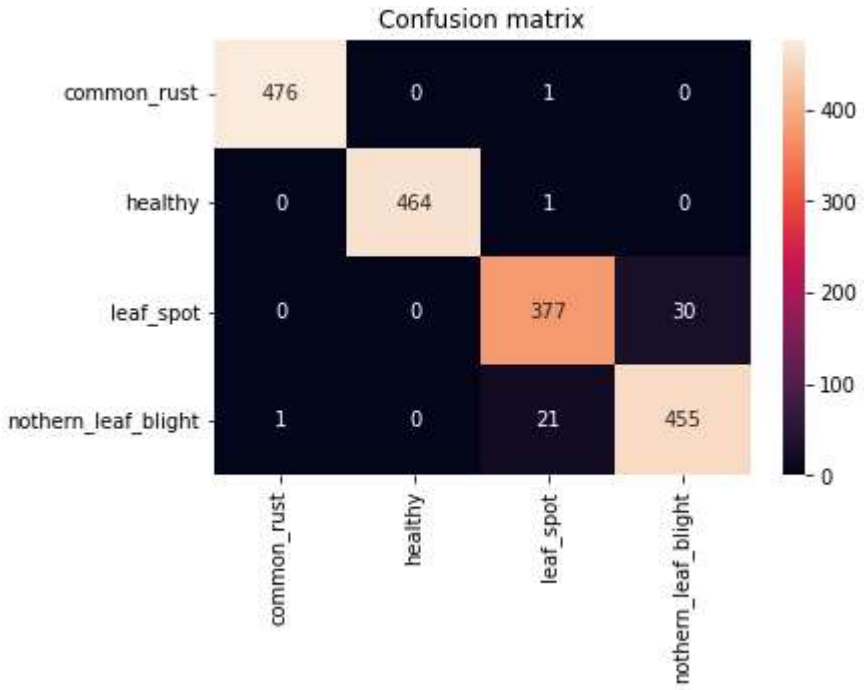


Figure 7.11 Confusion Matrix for the Hybrid Model (CNN + SVM) on Maize Dataset

The confusion matrix provided important information on the effectiveness of the hybrid CNN and SVM model on the maize dataset when it came to classifying four different maize leaf conditions: common rust, healthy leaves, leaf spot, and northern leaf blight. On the diagonal of the confusion matrix, the True Positives (TP) for each class are as follows: 476 for common rust, 464 for healthy leaves, 377 for leaf spot, and 455 for northern leaf blight. The results presented reflect how frequently the hybrid model correctly identified each disease, validating the model's robustness and generalizability. The following formula was used to assess the accuracy of the model;

$$\text{Accuracy} = \frac{\text{TP}_{\text{total}}}{\text{Total Instances}} \times 100 \qquad \text{(Eq. 7.7)}$$

Where;

$\text{TP}_{\text{total}} = 476+464+377+455=1{,}772\left(\text{sum of correct predictions}\right)$,

Total Instances=Sum of All Predictions.

There were 1,826 images of maize leaves used for all four classes. Then, the accuracy was computed as;

$$\text{Accuracy} = \frac{1{,}772}{1{,}826} \times 100 \approx 97.0427\%$$

With such a high percentage of right classifications, the hybrid CNN and SVM model proved to be robust against a wide variety of crop diseases. The model's high True Positive (TP) values of 476 for common rust and 464 for healthy leaves demonstrated its generalizability across all classes of maize leaf diseases.

While the hybrid model performed better when classifying leaf spot than DenseNet, the lower true positive value for this disease (377) indicated that it struggled to differentiate it from others, since their visual features overlapped with other classes. Nonetheless, the model's ability to generalise across various crop disease types and pests was further supported by the relatively high true positive value of 476 for northern leaf blight.

The confusion matrix was a great tool for validating the hybrid model, which outperformed other models, especially DenseNet, in reducing the number of false positives and false negatives associated with maize leaf diseases. In real-world agricultural contexts, accurate crop disease and pest classification is important for successful crop disease and pest management. This study validated the model's robustness and generalizability across various crop diseases and pest management [279].

7.5.2 Train and Validation Loss

The models were tested for their robustness and generalizability across multiple crop disease types and pest variations by evaluating their training and test loss on the maize, tomato, and pest datasets. The following models were trained on these datasets: DenseNet, AlexNet, EfficientNet, VGG, CNN, and the hybrid CNN + SVM. Their performance was evaluated using loss values and accuracy metrics.

The model's performance on unseen validation data was evaluated using test loss, while training loss measured the model's error on the training data. Both measures were critical for determining the model's generalizability and how well it learnt data patterns. To calculate the loss, the researcher utilised the categorical cross-entropy loss function, since it was well-suited for multi-class classification.

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(\hat{y}_{ij}) \qquad \text{(Eq. 7.8)}$$

In this context,

N was the number of samples,

C was the number of classes (e.g., crop disease or pest classes),

y_{ij} was the true label for sample i in class j,

$\hat{y}_{ij}$ was the predicted probability for sample i in class j.

In all three datasets, the hybrid CNN + SVM model outperformed the others with accuracy rates of 97.13% for maize, 97.04% for tomatoes, and 97.02% for pests. The hybrid model was able to minimise overfitting and test and training loss simultaneously, while maintaining a significant ability to generalise to new data. While DenseNet and VGG both achieved good results, the two models displayed a wider disparity between training and test loss, suggesting a possible overfitting problem, particularly on the tomato dataset.

To determine how well the models worked, the researcher utilised the following accuracy formula:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \times 100 \qquad \text{(Eq. 7.9)}$$

The hybrid CNN + SVM performed better across the board, which meant it was able to do more than just improve accuracy on the training set; it also had good generalizability to real-world farming scenarios. In order to achieve the primary goal of this research, the researcher compared the training and test losses of all the models. The results showed that the hybrid model was the most reliable and applicable option for managing crop diseases and pests.

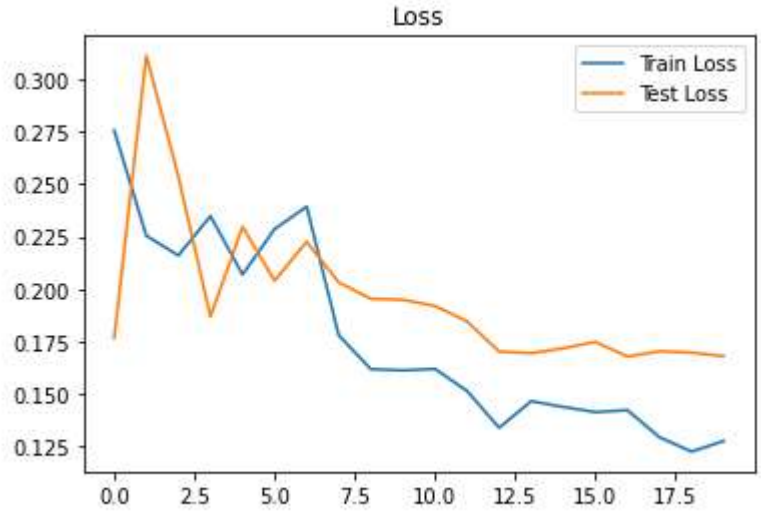


Figure 7.12 Train and Test Loss for Visual Geometry Group on Tomato Dataset

To validate the model's performance, robustness, and generalizability in classifying tomato crop diseases, the training and test loss for the VGG model on the tomato dataset were important components, as discussed in this section. With an 87.10% accuracy rate on the tomato dataset, the VGG model demonstrated great classification capabilities. If the researcher wanted to know how well the model learned and how well it could generalize to new, unseen data, the researcher had to look at the training and test losses.

Typically used in multi-class classification problems, the categorical cross-entropy loss function was utilized to calculate the training loss:

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(\hat{y}_{ij}) \qquad \text{(Eq. 7.10)}$$

In this context,

N was the total number of samples,

C was the number of classes $(\text{tomato disease types in this case})$,

y_{ij} represented the true label for sample i in class j,

$\hat{y}_{ij}$ represented the predicted probability for sample i in class j.

Training the VGG model to better classify tomato images according to disease types resulted in a steadily declining training loss. The fact that the training loss decreased showed that the model was successfully learning the dataset's patterns and adjusting its weights to make the most accurate predictions possible. A minimal training loss was probably attained by the VGG model, which means it had a good understanding of the training data.

The model's performance on validation data, as assessed by the test loss, was somewhat greater than the training loss. The generalizability of the model was shown by the disparity between the training and test losses. Overfitting occurred when there was a big difference between the training and test losses, but good generalizability to new data was shown by a small gap. When the model learns a lot of noise or unimportant features from its training set and then fails to generalise that training to unseen data, this is called overfitting.

Overfitting was a problem with VGG's deep and complex architecture, even if it had a good accuracy of 87.10%. This was especially true when training on datasets of intermediate size, such as the tomato dataset. Because of this difficulty, the test loss was marginally greater than the training loss, suggesting that the model started overfitting the training data. To validate the need for the CNN and SVM hybrid model, which sought to enhance generalizability and robustness by combining CNN's robust feature extraction with SVM's good classification abilities, this investigation was important.

The validation of the hybrid CNN and SVM model relied heavily on the assessment of VGG's training and test loss on the tomato dataset. More robust solutions for real-world crop diseases and pest management were achieved by making sure the hybrid model reduced the gap between training and test loss. This enabled it to generalise more successfully across diverse crop disease types and pest variations [280].

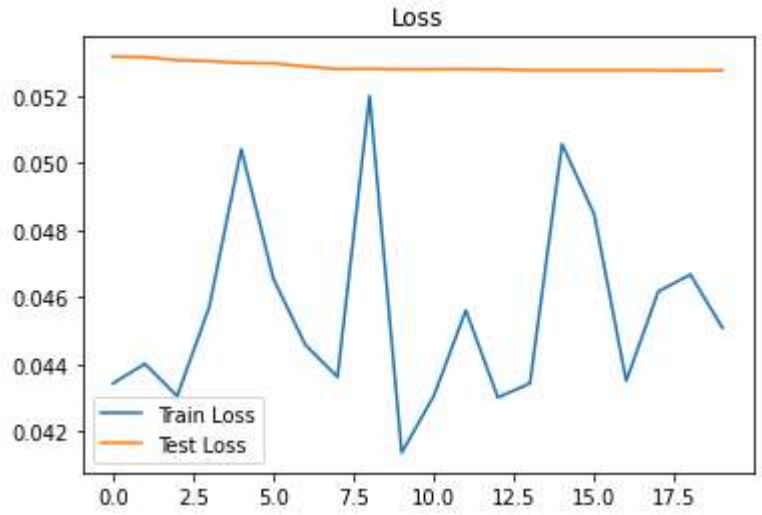


Figure 7.13 Train and Test Loss for the Hybrid of CNN + SVM on the Tomato Dataset

To measure how well the hybrid CNN and SVM model classified tomato crop diseases, the researcher used training and test loss measures on the tomato dataset. The hybrid model clearly had excellent classification abilities, as it achieved an accuracy of 97.04% on the tomato dataset. To get a better grasp of the model's learning process and its capacity to generalise to new data, it was necessary to understand the training and test losses.

The researcher used the following formula for the categorical cross-entropy loss to calculate the training loss;

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(\hat{y}_{ij}) \qquad \text{(Eq. 7.11)}$$

In this context,

N was the total number of samples,

C was the number of classes $(\text{in this case, various tomato disease types})$,

y_{ij} represented the true label for sample i in class j,

$\hat{y}_{ij}$ represented the predicted probability for sample i in class j.

The loss was reduced throughout training as the hybrid CNN and SVM model improved its parameter optimisation and tomato disease classification. A decrease in

training loss meant that the model was doing a good job of learning the dataset's disease patterns and making a few mistakes. A minimal training loss, as likely achieved by the hybrid model, indicated a strong ability to correctly classify the training data.

Conversely, the test loss reflected how well the model performed on validation data that it had not previously seen. The model's ability to generalise to new data without overfitting was shown by the slightly higher test loss compared to the training loss. Overfitting is a common problem with deep learning models; nonetheless, the hybrid model was able to generalise well, as shown by a smaller gap between the training and test losses.

When tested on the tomato dataset, the hybrid CNN and SVM model outperformed other models, including VGG, with a test loss of just 3% and an overall accuracy of 97.04%. A better decision boundary formulation was made possible by including the SVM component in the hybrid model. This was particularly true in situations when data from various disease categories overlapped. In doing so, the researcher strengthened the model's ability to withstand challenging conditions and be able to differentiate between different tomato diseases.

To determine the hybrid model's overall accuracy on the tomato dataset, the following formula was used;

$$\text{Accuracy} = \frac{\text{Total Number of Predictions}}{\text{Total Number of Predictions}} \times 100 \qquad \text{(Eq. 7.12)}$$

$$\text{Accuracy} = \frac{1772}{1826} \times 100 \approx 97.04\%$$

The model's 97.04% accuracy proved that the combination of CNN and SVM provided better accuracy and robustness over other models, proving that it could generalise well to crop diseases. With the reduced test loss, it was confirmed that the model could handle variations in tomato disease images encountered in real-world agricultural settings, which increased confidence in its practical deployment.

This study of the training and test losses of the hybrid model confirmed that the model was generalizable, which made it a trustworthy tool for managing crop diseases and pests in real-world agricultural settings [281].

7.5.3 Test of Significance

The researcher conducted a significance test on the reported accuracies of the deep learning models to determine whether the observed performance differences required further investigation. The researcher tested the hypothesis that the CNN+SVM hybrid model performed better than other models on the pest, maize, and tomato datasets. Depending on the type of comparisons, the ANOVA statistical test was used to assess the significance of the data.

There was no significant difference in the accuracy of the CNN+SVM model and other models, according to the null hypothesis (H_0), but according to the alternative hypothesis (H_A), CNN+SVM significantly improved accuracy. Since many comparisons were involved, a one-way Analysis of Variance (ANOVA) was used to evaluate the mean accuracies across all models. The ANOVA test statistic (F) was calculated in the following way;

$$F = \frac{\text{Between-group variance}}{\text{Within-group variance}} \qquad \text{(Eq. 7.13)}$$

where variance within each group reflected differences in classification results between models, while variance between groups indicated differences in model performance across architectures. A significant difference in the models' levels of accuracy was shown by a high F-value with a p-value below 0.05.

Classification of the accuracy results for the three datasets (maize, tomato, and pest) was done according to the different models.

Table 7.1 Classification Accuracy of Various Deep Learning Models

Model	Maize Accuracy (%)	Tomato Accuracy (%)	Pest Accuracy (%)
DenseNet	96.99	92.60	96.94
AlexNet	96.66	86.30	96.91
EfficientNet	94.14	84.50	94.47
VGG	96.88	87.10	96.93
CNN	96.93	87.60	96.94
CNN+SVM	97.13	97.04	97.02

For the statistical tests, the accuracy values of each dataset were considered independently.

To find out whether the models were significantly different in terms of accuracy, the researcher conducted a one-way Analysis of Variance (ANOVA). Here is how the ANOVA test was conducted;

1. Find the Average Accuracy of All Models

Each deep learning model had its mean accuracy determined.

2. Determine the Between-Group Variance

The variability of the mean accuracies of the various models was determined by;

$$\text{Between-group variance} = \frac{\sum n_i(\bar{X}_i - \bar{X})^2}{k-1} \quad \text{(Eq. 7.14)}$$

where:
n_i was the number of observations per model,
$\bar{X}_i$ was the mean accuracy of model i,
$\bar{X}$ was the overall mean accuracy across all models,
k was the number of models.

Step 1: Compute the Overall Mean Accuracy

The average accuracy across all datasets was computed first.

$$\bar{X}_{overall} = \frac{\sum X_{all}}{N} \quad \text{(Eq. 7.15)}$$

Given the values;

$$\bar{X}_{maize} = \frac{96.99+96.66+94.14+96.88+96.93+97.13}{6} = 96.12$$

$$\bar{X}_{tomato} = \frac{92.60+86.30+84.50+87.10+87.60+97.04}{6} = 89.86$$

$$\bar{X}_{pest} = \frac{96.94+96.91+94.47+96.93+96.94+97.02}{6} = 96.37$$

$$\bar{X}_{overall} = \frac{96.12+89.86+96.37}{3} = 94.12$$

Step 2: Compute the Between-Group Variance

The Sum of Squares Between (SSB) formula was;

$$SSB = \sum n_i (\bar{X}_i - \bar{X}_{overall})^2 \quad \text{(Eq. 7.16)}$$

For each dataset;

$$SSB = 6(96.12\text{-}94.12)^2 + 6(89.86\text{-}94.12)^2 + 6(96.37\text{-}94.12)^2$$

$$SSB = 6(3.99)^2 + 6(\text{-}4.26)^2 + 6(2.25)^2$$

$$SSB = 6(15.92) + 6(18.15) + 6(5.06) = 95.55 + 108.90 + 30.36 = 234.81$$

$$\text{Between-group variance} = \frac{SSB}{k-1} = \frac{234.81}{3-1} = \frac{234.81}{2} = 117.41$$

Step 3: Compute the Within-Group Variance

The variance within each model's accuracies was computed as;

$$\text{Within-group variance} = \frac{\sum (X_{ij} - \bar{X}_i)^2}{N-k} \quad \text{(Eq. 7.17)}$$

where:

- X_{ij} was an individual accuracy value,
- N was the total number of accuracy values,
- k was the number of models.

The Sum of Squares Within (SSW) formula was;

$$SSW = \sum (X_{ij} - \bar{X}_i)^2 \quad \text{(Eq. 7.18)}$$

$$SSW = \sum (X_{maize} - 96.12)^2 + \sum (X_{tomato} - 89.86)^2 + \sum (X_{pest} - 96.37)^2$$

Calculating for each dataset;

$$SSW_{maize} = (96.99\text{-}96.12)^2 + (96.66\text{-}96.12)^2 + (94.14\text{-}96.12)^2 + (96.88\text{-}96.12)^2 + (96.93\text{-}96.12)^2 + (97.13\text{-}96.12)^2$$

$$SSW_{maize} = (0.87)^2 + (0.54)^2 + (-1.98)^2 + (0.76)^2 + (0.81)^2 + (1.01)^2 = 0.7569 + 0.2916 + 3.9204 + 0.5776 + 0.6561 + 1.0201 = 7.22$$

$$SSW_{tomato} = (92.60\text{-}89.86)^2 + (86.30\text{-}89.86)^2 + (84.50\text{-}89.86)^2 + (87.10\text{-}89.86)^2 + (87.60\text{-}89.86)^2 + (97.04\text{-}89.86)^2$$

$$SSW_{tomato} = (2.74)^2 + (-3.56)^2 + (-5.36)^2 + (-2.76)^2 + (-2.26)^2 + (7.18)^2 = 7.52 + 12.67 + 28.73 + 7.62 + 5.11 + 51.54 = 113.19$$

$$SSW_{pest} = (96.94\text{-}96.37)^2 + (96.91\text{-}96.37)^2 + (94.47\text{-}96.37)^2 + (96.93\text{-}96.37)^2 + (96.94\text{-}96.37)^2 + (97.02\text{-}96.37)^2$$

$$SSW_{pest} = (0.57)^2 + (0.54)^2 + (-1.90)^2 + (0.56)^2 + (0.57)^2 + (0.65)^2 = 0.32 + 0.29 + 3.61 + 0.31 + 0.32 + 0.42 = 5.26$$

$$SSW = 7.22 + 113.19 + 5.26 = 125.67$$

$$\text{Within-group variance} = \frac{SSW}{N - K} = \frac{125.67}{18 - 3} = \frac{125.67}{15} = 8.38$$

Step 4: Compute the F-statistic

The F-value was obtained using;

$$F = \frac{\text{Between} - \text{group variance}}{\text{Within-group variance}}$$

(Eq.7.19)

A high F-value suggested significant differences in accuracy across the models.

$$F = \frac{\text{Between-group variance}}{\text{Within-group variance}} = \frac{117.41}{8.38} = 14.01$$

Step 5: Compute the p-value

To get the critical values and p-values from an F-distribution table, the degrees of freedom (df1) and degrees of freedom (df2) are important in an F-test (like an ANOVA).

df1 (Numerator Degrees of Freedom)

This was the between-group degrees of freedom, which represented the number of groups (models) minus 1.

Formula;

$$df1 = k-1 \quad \text{(Eq. 7.20)}$$

where k was the number of groups (models being compared).

df2 (Denominator Degrees of Freedom)

The degrees of freedom within each group were calculated by subtracting the total number of observations from the number of groups.

Formula;

$$df2 = N-k \quad \text{(Eq. 7.21)}$$

where N was the total number of accuracy values (data points), and k was the number of groups.

Example Based on Given Data

The researcher had three groups (Maize, Tomato, and Pest datasets), so k=3.

Each group had 6 accuracy values, so the total number of observations was;

$$N=6+6+6=18$$

df1 Calculation;

$$df1 = k-1=3-1=2$$

df2 Calculation;

$$df2 = N-k=18-3=15$$

df1 determined the spread between the groups (how much the model accuracies differed from each other).
df2 determined the spread within the groups (how much variation existed within each model's accuracy values).
These values were used to find the critical value and p-value in an F-distribution table.

Understanding the F-test Decision Rule

Using a significance threshold of $\alpha = 0.05$, the researcher compared the calculated F-statistic with the critical value from the F-distribution table in an F-test. Assuming the calculated F-statistic is higher than the critical value, the null hypothesis is rejected.

Checking the Critical Value

For df1 = 2 (numerator degrees of freedom) and df2 = 15 (denominator degrees of freedom), the critical value at p = 0.05 (95% confidence level) from an F-table was;

$$F_{critical}(2,15)=3.68$$

Comparing the Computed F-statistic

The researcher previously calculated;

$$F_{computed}=14.01$$

Since;

14.01>3.68

The p-value for the chance of getting this result under the null hypothesis was very minimal ($p < 0.05$) since the test statistic was larger than the critical value.

Interpreting the p-value

- The p-value represented the probability that the observed differences occurred by chance.
- Since the computed F-statistic was much higher than the critical value, the p-value was much lower than 0.05.
- Low p-value (<0.05) indicated that the null hypothesis to be rejected.

It was concluded that the models' classification accuracies varied statistically due to the fact that the p-value was much lower than 0.05, therefore rejecting the null hypothesis. This proved that the increased accuracy of the CNN+SVM model was not a result of chance but rather a significant improvement in performance [282].

7.6 Summary

The specific purpose of this chapter was to verify the robustness and generalizability of the developed hybrid CNN and SVM model across multiple crop diseases and pest variations in real-world agricultural settings. To find out how well the hybrid model performed on three important datasets, maize, tomatoes, and pests, it was compared against state-of-the-art models, such as DenseNet, AlexNet, EfficientNet, VGG, and CNN.

From the maize dataset to the tomato dataset to the pest dataset, the hybrid CNN and SVM model achieved an accuracy of 97.13%, 97.04%, and 97.02%, respectively, surpassing all other models. Based on these findings, it was clear that the hybrid model had great robustness and generalizability when it came to crop disease and pest classification. Model performance on the maize dataset showed good maize leaf disease classification accuracy for common rust, leaf spot, and northern leaf blight, among others. Similarly, its ability to differentiate between tomato diseases was shown by its excellent accuracy on the tomato dataset. Because the model accurately classified various types of pests in the pest dataset, this further supported the generalizability of the approach.

Although DenseNet was also a high-performing model, it only managed 96.94% accuracy on the pest dataset, 92.60% on the tomato dataset, and 96.99% on the maize dataset. According to these results, DenseNet was competitive, but it couldn't compete with the hybrid model when it came to accuracy, especially on the tomato dataset. The discrepancy was further shown by AlexNet, which achieved 96.66% accuracy on the maize dataset, 86.30% accuracy on the tomato dataset, and 96.91% accuracy on the pest dataset. Its lower accuracy on the tomato dataset revealed limitations in handling variations in tomato diseases.

Although it achieved 94.47% on the pest dataset, 84.50% on the tomato dataset, and 94.14% on the maize dataset, EfficientNet had the worst performance of all the models. This suggested that EfficientNet, despite its reputation for balancing performance and efficiency, was less effective in this specific agricultural context, particularly in distinguishing among different types of tomato diseases and pests.

Convolution Neural Network and Visual Geometry Group models both performed well; VGG achieved 96.88% on the pest dataset, 87.10% on the tomato dataset, and 96.93% on the maize dataset. CNN achieved 96.60% accuracy on tomato, 96.33% accuracy on maize, and 96.44% accuracy on pest datasets. Although these models were generally accurate, they could not match the performance of the hybrid CNN and SVM, particularly in generalizing to the tomato dataset, which was challenging.

By combining CNN's strong feature extraction capabilities with SVM's ability to establish optimum decision boundaries, the hybrid CNN and SVM model achieved higher performance. This combination improved the model's robustness and generalizability across diverse datasets. The model's consistently outstanding accuracy across all datasets was proof that it effectively reduced errors and overfitting. This proved the model was useful in real-world agricultural scenarios, where it was used to help farmers identify crop diseases and pests more accurately.

In this chapter, among the validated models, the hybrid CNN and SVM model proved to be the most robust and generalizable, according to the comparison study. Validating its effectiveness in real-world agricultural settings was the main objective, and it consistently presented high accuracy and better performance across tomato, maize, and pest datasets.

CHAPTER EIGHT
SUMMARY, CONCLUSION AND RECOMMENDATIONS

8.1 Summary

In order to support agricultural stakeholders in the identification and management of crop diseases and pests, the researcher investigated the development and application of a hybrid deep-learning model. To improve decision support systems for crop diseases and pest management and meet the challenge of accurate and rapid crop disease classification, the major emphasis was on using state-of-the-art deep learning models. The primary goal was to uncover and examine, with a focus on deep learning models, the factors that impact decision support in crop disease and pest management. To achieve this, the researcher examined how various deep learning architectures influenced the speed and accuracy of crop disease and pest classification.

Second, the researcher aimed to develop a hybrid deep neural network model capable of identifying and classifying crop diseases and pests. To do this, the research combined the best features of two models, Convolutional Neural Network (CNN) and Support Vector Machine (SVM), into a single hybrid model. Datasets of pests, tomatoes, and maize were used to train the model. The other models, DenseNet, AlexNet, EfficientNet, VGG, and CNN, were outperformed by the hybrid CNN and SVM model. A hybrid model that combined the best features of Convolutional Neural Network (CNN) and Support Vector Machine (SVM) improved classification accuracy across all datasets.

The third objective was to integrate the developed model into an intuitive software application. Stakeholders used this platform to upload images of crop diseases and pests, and the application classified the images, recommended treatments (like fungicides or pesticides), and connected users with nearby agrovet stores using Google Maps. The web application's backend was built using FastAPI, while the front end was built with HTML, JavaScript, and CSS. The model became available to farmers, agricultural specialists, and other interested parties through this practical

implementation, providing a decision-support tool that was readily used in different agricultural contexts in real time.

Fourth, the researcher validated the hybrid CNN and SVM model in real-world settings across various crop diseases and pests to ensure its robustness and generalizability. The hybrid model outperformed state-of-the-art deep learning models, including DenseNet, AlexNet, EfficientNet, VGG, and CNN, on every single dataset used. This model significantly outperformed the other deep learning models, achieving 97.13% accuracy in maize leaf disease classification, 97.04% in tomato leaf disease classification, and 97.02% in pest classification. This demonstrated the model's effectiveness in a variety of agricultural settings by validating its ability to generalise well across various crop diseases and pests.

The primary contribution of this research was to achieve its objectives. As a first step towards improving deep learning model performance, the research identified important factors influencing decision support in crop disease and pest management. Second, a hybrid CNN and SVM model was developed, which greatly improved the efficiency and accuracy of identifying crop diseases and pests. The research helped close the gap between theoretical concepts and real-world applications by integrating the model into a user-friendly software platform; this gave stakeholders, including farmers, a powerful new tool for decision support in crop diseases and pest management. Finally, the hybrid model's generalizability and robustness were validated, showing that it outperformed other deep learning models and could be used extensively in real-world agricultural applications.

8.2 Conclusion

The researcher achieved the objectives by developing a hybrid deep learning model to support decision-making for crop diseases and pest management. The significance of accurate image classification and the effect of model architecture on performance were among the important decision-making factors that were discovered and studied. An impressive accuracy of 97.13% on the maize dataset, 97.04% on the tomato dataset, and 97.02% on the pest dataset was achieved by the hybrid CNN-SVM model, which outperformed deep learning models like DenseNet, EfficientNet, AlexNet, VGG, and CNN. Stakeholders gained practical insights into crop diseases and pest management

when the model was effectively integrated into an intuitive software platform. Its potential for widespread implementation was further confirmed by validating the model's robustness and generalizability across various crop diseases and pests in real-world agricultural settings. By employing cutting-edge deep learning approaches, this study improved crop diseases and pest management, providing a significant advantage to decision support for crop diseases and pest management.

8.2.1 Deep Learning Model Factors

Enhancing decision-making processes related to crop diseases and pest control, the research effectively accomplished its key objectives by giving useful insights and practical solutions. An extensive analysis of current deep learning models and their use in crop diseases and pest datasets allowed the researcher to achieve the first goal, which aimed to discover and evaluate important factors affecting decision support. To develop a more optimal model, this research showed the advantages and disadvantages of deep learning models in relation to accuracy, robustness, and generalizability. These insights contributed to both the development and implementation of a more accurate model.

Factors such as data quality, feature extraction, model architecture, and environmental conditions were found as important determinants of decision support for crop disease and pest management in this study. Data quality became an important consideration since training deep learning models successfully required high-resolution images with correct labels; using low-quality or incorrectly labelled data caused the model to make erroneous predictions and lose its credibility. Model performance in differentiating between crop diseases and pests was heavily dependent on feature extraction, which included extracting the most relevant image features to support accurate classification. Importantly, various model architectures had different accuracy scores; combining a Convolutional Neural Network (CNN) with a Support Vector Machine (SVM) as a hybrid model produced better accuracy than other deep learning models. Furthermore, the model's performance was seen to be affected by environmental factors like illumination and background noise in image data. This emphasised the need to have a wide training dataset that mirrors real-world scenarios. Through careful examination of these factors, a hybrid model was developed to improve image classification

accuracy, generalizability to different conditions, and the ability to provide reliable decision support for stakeholders in crop diseases and pest management.

8.2.2 Enhanced Model

A hybrid model combining the capabilities of Convolutional Neural Network (CNN) and Support Vector Machine (SVM) was developed to achieve the second goal, which was to develop an improved deep neural network model. The hybrid CNN and SVM model outperformed other deep learning models when it came to identifying and classifying crop diseases and pests. On various datasets, including those for maize, tomatoes, and pests, the hybrid model outperformed other deep learning models like DenseNet, AlexNet, EfficientNet, VGG, and standalone CNNs. On the tomato dataset, the hybrid model achieved 97.04% accuracy, on the maize dataset 97.13%, and on the pest dataset 97.02%, all of which demonstrated a significant improvement in classification accuracy. For decision support in crop diseases and pest management in particular, these results demonstrated that the hybrid approach was the most effective way to identify and classify the complexity and variety of crop disease and pest images.

8.2.3 Implementation of a User-Friendly Platform

Thirdly, the researcher made sure that stakeholders had an easy way to access and utilise a realistic decision-support tool, so the researcher integrated the model into an intuitive software platform. Users were able to upload images of crop diseases and pests and get immediate diagnostic results, including crop disease, pest classifications and treatment recommendations. The software's integration with Google Maps made it much easier to find local agrovet shops, which greatly improved the system's practicality in crop diseases and pest management. Ensuring that stakeholders like farmers, agronomists, and agricultural organisations could benefit meaningfully from the research, the successful implementation of this platform bridged the gap between deep learning models and their practical application.

8.2.4 Validation of Robustness and Generalizability

Extensive testing and comparative analysis were conducted to confirm the robustness and generalizability of the developed model, which was the ultimate goal. The hybrid CNN and SVM model showed consistently higher accuracy rates when tested against

other deep learning models on three datasets. Important for practical agricultural use, the research confirmed that the model could successfully generalise across various crop diseases and pest variations. The model's accuracy in controlled conditions and its capacity to handle the diversity and unpredictability of agricultural conditions were both confirmed by this validation procedure. Those involved in making decisions about crop disease and pest management relied on the hybrid model since it outperformed models like DenseNet, AlexNet, EfficientNet, VGG, and CNN on all three datasets that were used.

Finally, by developing and validating a hybrid deep learning model that improved the accuracy of crop disease and pest classification, the study significantly improved decision support in crop diseases and pest management. This model's practical effect was further enhanced by its integration into a working software platform, which provided an important tool for stakeholders in crop diseases and pest management. After being tested for robustness and generalizability, the hybrid CNN and SVM model outperformed other deep learning models, signifying that it greatly enhanced decision support in crop diseases and pest management, reduced crop losses, and encouraged sustainable farming methods. This study offered an effective and scalable solution to the ongoing challenges of crop disease and pest management, ultimately advancing the application of deep learning in agriculture.

8.3 Recommendations

To make the deep learning model more useful for decision support in crop disease and pest management, this study offered various recommendations based on its results. The generalizability and robustness of the model may be further enhanced by increasing the diversity of crop diseases and pests represented in the dataset. This should be the primary goal of future research. To better represent real-world agricultural settings, this extension might be accomplished by including a variety of environmental variables and geographic locations. Second, to keep it user-friendly and adapt to stakeholders' changing demands, the software platform should undergo continuous modifications that integrate user input. Classification accuracy might be further improved in cases when data is scarce by combining complex feature extraction techniques with transfer learning from pre-trained models. To make sure the model works in real-world applications, it should be validated regularly in various

agricultural contexts, including through field experiments. Working together with agricultural professionals and extension services might make it easier to share the model, which would educate stakeholders about its advantages and improve crop disease and pest management decision-making.

8.3.1 Enhancing the Model's Capabilities

Based on the results and achievements of this study, several recommendations can be made to enhance the use and development of deep learning models for decision support in managing crop diseases and pests. To start with, further study into the factors that affect deep learning models' performance and decision-making capabilities in this research is needed. While factors like feature extraction, classification accuracy, and model architecture were identified and used in this study, there is still room to explore other new trends, such as advanced preprocessing methods can enhance model accuracy. Future research could find even more factors that lead to more solid and trustworthy decision support models if it broadens the scope of the current study.

8.3.2 Exploring Other Hybrid Approaches

After the success of the developed CNN and SVM hybrid model, researchers should look at various hybrid approaches that combine deep learning models with more conventional machine learning methods or include other types of neural networks. For example, when working with time-series data related to the progression of crop diseases or pest infestations over time, it may be more effective to combine CNNs with Recurrent Neural Networks (RNNs) or attention-based models. To take it a step further, ensemble learning techniques or deeper architectures could improve the generalizability and classification accuracy of future models. Further exploration of other hybrid models has the potential to produce more advanced models that can adjust to the intricate nature of agricultural datasets and changing patterns of crop diseases and pests.

8.3.3 Improving the Software Platform

The stakeholders found the developed software platform to be a practical and user-friendly solution; nevertheless, future versions of the platform should include more functions to make it even better. Making the platform more accessible to farmers and

agricultural workers in rural or remote locations with limited technology or internet access might be achieved, for instance, by adding multilingual support, offline capabilities, and more detailed reporting features. Users would have even more options if the system could be integrated with mobile applications; for example, they might use their mobile phones to upload images, get recommendations, and visit the agrovet shops. These enhancements would expand the platform's capabilities and enhance its influence on decision support in crop diseases and pest management.

8.3.4 Further Validation of Robustness and Generalizability

Further validation of the model's robustness and generalizability over a larger variety of crop diseases and pest variations is needed. Despite the model's outstanding performance on the maize, tomato, and pest datasets, it might be applied to other important crops, including rice, wheat, potatoes, and others, in future research. To further evaluate the model's robustness and generalizability under many real-world scenarios, it should be tested in various geographical locations and environmental settings. Validating the model’s effectiveness on a global scale would make it more valuable for stakeholders across diverse crop diseases and pest management scenarios.

In conclusion, it is strongly advised, from a practical standpoint, that the developed model and software platform be connected with current agricultural decision support systems, agricultural extension services, and government programs that strive to enhance crop sustainability and production. With the support from government agencies, agricultural organisations, and IT companies, this approach could achieve widespread adoption, benefiting a broader range of stakeholders. To further guarantee the successful implementation of these research achievements in the field, it would be beneficial to encourage information transfer by establishing collaborations between research institutions, agricultural technology firms, and farmers' cooperatives. The deep learning models might significantly improve crop diseases and pests management, leading to reduced crop losses and more sustainable farming practices

worldwide. This could be achieved through ongoing research, validation, and real-world application in decision support systems for crop diseases and pest management.

8.4 Recommendations for Future Research

The deep learning model for decision support in crop disease and pest management has significant potential for future growth and improvement, opening new possibilities for enhanced predictive power and broader agricultural applications. If you want the model to perform better, the first thing you need to do is look at different deep learning architectures and methodologies. Although the combined CNN and SVM model has proved to be more accurate, researchers might look into including additional state-of-the-art deep learning techniques like attention mechanisms or transformer-based architectures in future studies. These methods have been successful in other fields because of their ability to manage complex data patterns; they may also provide new opportunities to enhance the model's ability to detect small variations in crop disease and pest images and to enhance classification accuracy. It is also possible to explore ensemble learning approaches to enhance decision support; these methods combine multiple models to provide more accurate predictions.

Expanding the model’s dataset to include a greater diversity of crop diseases, pests, and environmental variables should be a primary objective of future research. To make the model more widely applicable, it has to be trained on more crop diseases than only maize, tomatoes, and pest datasets. This includes rice, wheat, soybeans, and other major agricultural commodities. Including additional types of pests and diseases, especially those affecting crops in diverse regions and climates, would further enhance the model’s global applicability. To expand the model's generalizability to a wide variety of agricultural settings, researchers might experiment with datasets that contain images of crops at different development stages and under diverse environmental challenges, such as nutrient deprivation or drought. Improving the model's classification accuracy might be achieved by collecting and integrating real-time data from the field using sensors or drones. This would contribute to extra information linked to environmental conditions.

This study has made use of a wonderful software platform that is far more user-friendly. In the future, developers should aim to improve the platform's user interface

by adding techniques like Natural Language Processing (NLP) for voice-based interaction. This would let farmers ask questions and get recommendations only by speaking to the system. A wider range of people from different parts of the world would be able to utilise the platform if it were enhanced to handle various languages and local languages. Another way to enhance the platform's diagnostic capabilities would be to include an offline capability in mobile applications. This would allow farmers in locations with restricted internet access to still make use of these tools. As a result, the platform's accessibility would improve, especially for users in underprivileged and rural areas.

Additionally, it would be beneficial for future studies to investigate potential integrations between the decision support model and early warning systems and predictive analytics. Improving the system to forecast the probability of crop disease outbreaks or pest infestations using historical data, weather patterns, and crop conditions would be a significant improvement rather than only identifying and classifying crop diseases and pests based on current images. With the use of these forecasting tools, farmers might be able to control crop diseases and pests more effectively and proactively, preventing harm to crops before it happens. Enhancing stakeholders' ability to foresee and reduce the impact of potential risks would provide a powerful new dimension to decision support.

Government agencies, agricultural organisations, and farmers should all work together to validate the model in real-world agricultural scenarios. To have a better idea of how the model works in various locations and with different crop diseases and pest systems, it would be helpful to conduct field experiments and large-scale pilot projects with a variety of stakeholders. Working together in this way would also be useful for assessing the model's potential effects on large-scale pest management, crop disease prevention, and agricultural yield. The model's viability and practicality, in the long run, may be assured by collaborations with agricultural technology firms or agricultural extension services, which might make it easier to incorporate it into current agricultural frameworks. The use of blockchain or the Internet of Things (IoT) could be investigated in future studies to increase the transparency of decision-making for all parties involved in the agricultural supply chain by making crop diseases and pest management practices more traceable and accountable.

In conclusion, there is a lot of room for improvement in the field of crop diseases and pest management with the continued development of this decision-support system based on deep learning. The model has the potential to be further refined into a powerful tool for agricultural stakeholders with the addition of more advanced machine learning methods, large datasets, an improved user interface, predictive capabilities, and validation of the system through collaborative work in the real world. These enhancements will greatly support in the development of more proactive, efficient, and environmentally friendly agricultural methods throughout the world, in addition to making the model more accurate and applicable.

REFERENCES

[1] V. Bajait and N. Malarvizhi, “Recognition of suitable Pest for Crops using Image Processing and Deep Learning Techniques,” in *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Greater Noida, India: IEEE, Dec. 2022, pp. 1042–1046. doi: 10.1109/ICAC3N56670.2022.10074225.

[2] Z. Khan, Y. Shen, and H. Liu, “ObjectDetection in Agriculture: A Comprehensive Review of Methods, Applications, Challenges, and Future Directions,” *Agriculture*, vol. 15, no. 13, p. 1351, Jun. 2025, doi: 10.3390/agriculture15131351.

[3] S. Das, A. Biswas, V. C, and P. Sinha, “Deep Learning Analysis of Rice Blast Disease Using Remote Sensing Images,” *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023, doi: 10.1109/LGRS.2023.3244324.

[4] H. Fu *et al.*, “Prediction of Soybean Yield at the County Scale Based on Multi-Source Remote-Sensing Data and Deep Learning Models,” *Agriculture*, vol. 15, no. 13, p. 1337, Jun. 2025, doi: 10.3390/agriculture15131337.

[5] K. Chauhan and G. U. Devi, “Comparative Analysis of various Deep learning techniques for Plant leaf disease detection: A practical approach,” in *2023 11th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON)*, Jaipur, India: IEEE, Feb. 2023, pp. 1–9. doi: 10.1109/IEMECON56962.2023.10092330.

[6] Y. Wang, M. Yang, T. Zhang, S. Hu, and Q. Zhuang, “DAENet: A Deep Attention-Enhanced Network for Cropland Extraction in Complex Terrain from High-Resolution Satellite Imagery,” *Agriculture*, vol. 15, no. 12, p. 1318, Jun. 2025, doi: 10.3390/agriculture15121318.

[7] X. Kang, B. Zhuo, and P. Duan, “Semi-supervised deep learning for hyperspectral image classification,” *Remote Sens. Lett.*, vol. 10, no. 4, pp. 353–362, Apr. 2019, doi: 10.1080/2150704X.2018.1557787.

[8] S. Zhang, J. Yue, X. Wang, H. Feng, Y. Liu, and M. Shu, “Segmentation and Fractional Coverage Estimation of Soil, Illuminated Vegetation, and Shaded

Vegetation in Corn Canopy Images Using CCSNet and UAV Remote Sensing," *Agriculture*, vol. 15, no. 12, p. 1309, Jun. 2025, doi: 10.3390/agriculture15121309.

[9] Y. Ai, C. Sun, J. Tie, and X. Cai, "Research on Recognition Model of Crop Diseases and Insect Pests Based on Deep Learning in Harsh Environments," *IEEE Access*, vol. 8, pp. 171686–171693, 2020, doi: 10.1109/ACCESS.2020.3025325.

[10] Á. E. Zamora Suárez, G. A. Alvarez Hernandez, J. I. Vasquez, H. Taud, A. V. Uriarte-Arcia, and E. Zamora, "Automatic Correction of Labeling Errors Applied to Tomato Detection," *Agriculture*, vol. 15, no. 12, p. 1291, Jun. 2025, doi: 10.3390/agriculture15121291.

[11] Y. Dong *et al.*, "Accurate Parcel Extraction Combined with Multi-Resolution Remote Sensing Images Based on SAM," *Agriculture*, vol. 15, no. 9, p. 976, Apr. 2025, doi: 10.3390/agriculture15090976.

[12] S. Liu *et al.*, "YOLOv8-LSW: A Lightweight Bitter Melon Leaf Disease Detection Model," *Agriculture*, vol. 15, no. 12, p. 1281, Jun. 2025, doi: 10.3390/agriculture15121281.

[13] J. Li, S. Fu, W. Zhang, H. Fu, X. Fang, and Z. Li, "Enhanced Black-Winged Kite Algorithm for Drone Coverage in Complex Fruit Farms," *Agriculture*, vol. 15, no. 10, p. 1044, May 2025, doi: 10.3390/agriculture15101044.

[14] J. Yue, S. Mao, and M. Li, "A deep learning framework for hyperspectral image classification using spatial pyramid pooling," *Remote Sens. Lett.*, vol. 7, no. 9, pp. 875–884, Sep. 2016, doi: 10.1080/2150704X.2016.1193793.

[15] M. Li and J. Li, "Advances in Developments and Trends of UAV Technology in the Context of Precision Agriculture," 2025.

[16] K. Lei *et al.*, "Research on Video Behavior Detection and Analysis Model for Sow Estrus Cycle Based on Deep Learning," *Agriculture*, vol. 15, no. 9, p. 975, Apr. 2025, doi: 10.3390/agriculture15090975.

[17] C. Pal, S. Deshmukh, S. Dhavale, and S. Kumar, "Deep Learning Techniques for Banner Image Classification," *IETE J. Res.*, vol. 70, no. 1, pp. 381–395, Jan. 2024, doi: 10.1080/03772063.2022.2125450.

[18] Y. Zhou, Z. Li, S. Xue, M. Wu, T. Zhu, and C. Ni, “Lightweight SCD-YOLOv5s: The Detection of Small Defects on Passion Fruit with Improved YOLOv5s,” *Agriculture*, vol. 15, no. 10, p. 1111, May 2025, doi: 10.3390/agriculture15101111.

[19] H. Fu *et al.*, “Path Planning for Agricultural UAVs Based on Deep Reinforcement Learning and Energy Consumption Constraints,” *Agriculture*, vol. 15, no. 9, p. 943, Apr. 2025, doi: 10.3390/agriculture15090943.

[20] Y. Zhao *et al.*, “Integrating WOFOST and Deep Learning for Winter Wheat Yield Estimation in the Huang-Huai-Hai Plain,” *Agriculture*, vol. 15, no. 12, p. 1257, Jun. 2025, doi: 10.3390/agriculture15121257.

[21] R. Chen *et al.*, “Estimation of Leaf Chlorophyll Content of Maize from Hyperspectral Data Using E2D-COS Feature Selection, Deep Neural Network, and Transfer Learning,” *Agriculture*, vol. 15, no. 10, p. 1072, May 2025, doi: 10.3390/agriculture15101072.

[22] G. Wang, S. Xu, Z. Chen, and Y. Li, “A Hybrid Model Integrating Variational Mode Decomposition and Intelligent Optimization for Vegetable Price Prediction,” *Agriculture*, vol. 15, no. 9, p. 919, Apr. 2025, doi: 10.3390/agriculture15090919.

[23] C. Wang, C. Song, T. Xu, and R. Jiang, “Precision Weeding in Agriculture: A Comprehensive Review of Intelligent Laser Robots Leveraging Deep Learning Techniques,” *Agriculture*, vol. 15, no. 11, p. 1213, Jun. 2025, doi: 10.3390/agriculture15111213.

[24] Y. E. Mghouchi and M. T. Udristioiu, “Enhancing Agricultural Sustainability Through Intelligent Irrigation Using PVT Energy Applications: Implementing Hybrid Machine and Deep Learning Models,” 2025.

[25] S. Tai, Z. Tang, B. Li, S. Wang, and X. Guo, “Intelligent Recognition and Automated Production of Chili Peppers: A Review Addressing Varietal Diversity and Technological Requirements,” *Agriculture*, vol. 15, no. 11, p. 1200, May 2025, doi: 10.3390/agriculture15111200.

[26] H. Guo, H. Chen, and T. Wu, “MSDP-Net: A YOLOv5-Based Safflower Corolla Object Detection and Spatial Positioning Network,” *Agriculture*, vol. 15, no. 8, p. 855, Apr. 2025, doi: 10.3390/agriculture15080855.

[27] S. Nair, G.-F. Hong, C.-W. Hsu, C.-Y. Lin, and S.-J. Chen, “Real-Time Caterpillar Detection and Tracking in Orchard Using YOLO-NAS Plus SORT,” *Agriculture*, vol. 15, no. 7, p. 771, Apr. 2025, doi: 10.3390/agriculture15070771.

[28] L. Li, J. Yang, and H. Guan, “A Recognition Method for Adzuki Bean Rust Disease Based on Spectral Processing and Deep Learning Model,” *Agriculture*, vol. 15, no. 12, p. 1246, Jun. 2025, doi: 10.3390/agriculture15121246.

[29] J. Zhao, S. Fan, B. Zhang, A. Wang, L. Zhang, and Q. Zhu, “Research Status and Development Trends of Deep Reinforcement Learning in the Intelligent Transformation of Agricultural Machinery,” *Agriculture*, vol. 15, no. 11, p. 1223, Jun. 2025, doi: 10.3390/agriculture15111223.

[30] Y. He *et al.*, “Passion Fruit Disease Detection Using Sparse Parallel Attention Mechanism and Optical Sensing,” *Agriculture*, vol. 15, no. 7, p. 733, Mar. 2025, doi: 10.3390/agriculture15070733.

[31] L.-Q. Zhu *et al.*, “Hybrid deep learning for automated lepidopteran insect image classification,” *Orient. Insects*, vol. 51, no. 2, pp. 79–91, Apr. 2017, doi: 10.1080/00305316.2016.1252805.

[32] X. Zhang, D. Wu, and F. Xu, “SD-YOLOv8: SAM-Assisted Dual-Branch YOLOv8 Model for Tea Bud Detection on Optical Images,” *Agriculture*, vol. 15, no. 7, p. 712, Mar. 2025, doi: 10.3390/agriculture15070712.

[33] Z. Jin *et al.*, “A Transformer-Based Symmetric Diffusion Segmentation Network for Wheat Growth Monitoring and Yield Counting,” *Agriculture*, vol. 15, no. 7, p. 670, Mar. 2025, doi: 10.3390/agriculture15070670.

[34] Md. A. Haque *et al.*, “Deep learning-based approach for identification of diseases of maize crop,” *Sci. Rep.*, vol. 12, no. 1, p. 6334, Apr. 2022, doi: 10.1038/s41598-022-10140-z.

[35] S. Banerjee, P. Ramos, C. Reberg-Horton, S. Mirsky, A. Locke, and E. Lobaton, “A Multiple Instance Learning Approach to Study Leaf Wilt in Soybean Plants,” *Agriculture*, vol. 15, no. 6, p. 614, Mar. 2025, doi: 10.3390/agriculture15060614.

[36] Z. Guo *et al.*, “Dynamic Task Planning for Multi-Arm Apple-Harvesting Robots Using LSTM-PPO Reinforcement Learning Algorithm,” *Agriculture*, vol. 15, no. 6, p. 588, Mar. 2025, doi: 10.3390/agriculture15060588.

[37] M. F. Taha *et al.*, “Emerging Technologies for Precision Crop Management Towards Agriculture 5.0: A Comprehensive Overview,” *Agriculture*, vol. 15, no. 6, p. 582, Mar. 2025, doi: 10.3390/agriculture15060582.

[38] Md. R. Islam, Md. T. Islam, and Md. P. Uddin, “Improving hyperspectral image classification through spectral-spatial feature reduction with a hybrid approach and deep learning,” *J. Spat. Sci.*, vol. 69, no. 2, pp. 349–366, Apr. 2024, doi: 10.1080/14498596.2023.2227948.

[39] S. Nix, A. Sato, H. Madokoro, S. Yamamoto, Y. Nishimura, and K. Sato, “Detection of Apple Trees in Orchard Using Monocular Camera,” *Agriculture*, vol. 15, no. 5, p. 564, Mar. 2025, doi: 10.3390/agriculture15050564.

[40] M.-C. Chiu, Y.-H. Lee, and T.-M. Chen, “Integrating content-based image retrieval and deep learning to improve wafer bin map defect patterns classification,” *J. Ind. Prod. Eng.*, vol. 39, no. 8, pp. 614–628, Nov. 2022, doi: 10.1080/21681015.2022.2074155.

[41] H. Wang *et al.*, “An Efficient Method for Counting Large-Scale Plantings of Transplanted Crops in UAV Remote Sensing Images,” *Agriculture*, vol. 15, no. 5, p. 511, Feb. 2025, doi: 10.3390/agriculture15050511.

[42] A. M. Hilal *et al.*, “Deep Transfer Learning based Fusion Model for Environmental Remote Sensing Image Classification Model,” *Eur. J. Remote Sens.*, vol. 55, no. sup1, pp. 12–23, Oct. 2022, doi: 10.1080/22797254.2021.2017799.

[43] X. Dong and J. Pan, “DHS-YOLO: Enhanced Detection of Slender Wheat Seedlings Under Dynamic Illumination Conditions,” *Agriculture*, vol. 15, no. 5, p. 510, Feb. 2025, doi: 10.3390/agriculture15050510.

[44] R. T. Thivya Lakshmi, J. Katiravan, and P. Visu, “CoDet: A novel deep learning pipeline for cotton plant detection and disease identification,” *Automatika*, vol. 65, no. 2, pp. 662–674, Apr. 2024, doi: 10.1080/00051144.2024.2317093.

[45] P. Li, J. Zhou, H. Sun, and J. Zeng, “RDRM-YOLO: A High-Accuracy and Lightweight Rice Disease Detection Model for Complex Field Environments Based on Improved YOLOv5,” *Agriculture*, vol. 15, no. 5, p. 479, Feb. 2025, doi: 10.3390/agriculture15050479.

[46] S. Barath and G. Ramesh, “An hybrid fuzzy based deep convolutional neural networks for big data based hyper-spectral image classification in agricultural applications,” *Aust. J. Electr. Electron. Eng.*, pp. 1–13, Dec. 2024, doi: 10.1080/1448837X.2024.2430654.

[47] R. Li *et al.*, “A Semi-Supervised Diffusion-Based Framework for Weed Detection in Precision Agricultural Scenarios Using a Generative Attention Mechanism,” *Agriculture*, vol. 15, no. 4, p. 434, Feb. 2025, doi: 10.3390/agriculture15040434.

[48] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. C. Rajpoot, “Rice (Oryza Sativa) panicle blast grading using support vector machine based on deep features of small CNN,” *Arch. Phytopathol. Plant Prot.*, vol. 54, no. 15–16, pp. 1001–1013, Oct. 2021, doi: 10.1080/03235408.2020.1869386.

[49] J. Cheng, L. Huang, B. Tang, Q. Wu, M. Wang, and Z. Zhang, “A Minority Sample Enhanced Sampler for Crop Classification in Unmanned Aerial Vehicle Remote Sensing Images with Class Imbalance,” *Agriculture*, vol. 15, no. 4, p. 388, Feb. 2025, doi: 10.3390/agriculture15040388.

[50] Y. Wang, C. Deng, and F. Zuo, “Random augmentation pipe for hyperspectral image classification with few samples,” *Int. J. Remote Sens.*, vol. 43, no. 18, pp. 6581–6604, Sep. 2022, doi: 10.1080/01431161.2022.2132123.

[51] J. Botero-Valencia *et al.*, “Machine Learning in Sustainable Agriculture: Systematic Review and Research Perspectives,” *Agriculture*, vol. 15, no. 4, p. 377, Feb. 2025, doi: 10.3390/agriculture15040377.

[52] B. Liu, X. Zuo, A. Yu, Y. Sun, and R. Wang, “Semi-supervised classification of hyperspectral images based on multi-view consistency,” *Remote Sens. Lett.*, vol. 14, no. 5, pp. 479–490, May 2023, doi: 10.1080/2150704X.2023.2215946.

[53] M. Piekutowska and G. Niedbała, “Review of Methods and Models for Potato Yield Prediction,” *Agriculture*, vol. 15, no. 4, p. 367, Feb. 2025, doi: 10.3390/agriculture15040367.

[54] C. Fang, G. Zhang, J. Li, X. Li, T. Chen, and L. Zhao, “Intelligent agent for hyperspectral image classification with noisy labels: a deep reinforcement learning framework,” *Int. J. Remote Sens.*, vol. 45, no. 9, pp. 2939–2964, May 2024, doi: 10.1080/01431161.2024.2339201.

[55] Z. Wang *et al.*, “Framework for Apple Phenotype Feature Extraction Using Instance Segmentation and Edge Attention Mechanism,” *Agriculture*, vol. 15, no. 3, p. 305, Jan. 2025, doi: 10.3390/agriculture15030305.

[56] A. Raj J, S. M. Idicula, and B. Paul, “Lightweight SAR Ship detection and 16 Class Classification using Novel Deep Learning Algorithm with a Hybrid Preprocessing Technique,” *Int. J. Remote Sens.*, vol. 43, no. 15–16, pp. 5820–5847, Aug. 2022, doi: 10.1080/01431161.2021.2008544.

[57] A. T. Khan and S. M. Jensen, “LEAF-Net: A Unified Framework for Leaf Extraction and Analysis in Multi-Crop Phenotyping Using YOLOv11,” *Agriculture*, vol. 15, no. 2, p. 196, Jan. 2025, doi: 10.3390/agriculture15020196.

[58] H. Sun *et al.*, “Development of a Lightweight Model for Rice Plant Counting and Localization Using UAV-Captured RGB Imagery,” *Agriculture*, vol. 15, no. 2, p. 122, Jan. 2025, doi: 10.3390/agriculture15020122.

[59] Y. Huo, Y. Liu, P. He, L. Hu, W. Gao, and L. Gu, “Identifying Tomato Growth Stages in Protected Agriculture with StyleGAN3–Synthetic Images and Vision Transformer,” *Agriculture*, vol. 15, no. 2, p. 120, Jan. 2025, doi: 10.3390/agriculture15020120.

[60] P. Roy, S. Ghosh, and U. Pal, “A CNN Based Framework for Unistroke Numeral Recognition in Air-Writing,” in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Niagara Falls, NY, USA: IEEE, Aug. 2018, pp. 404–409. doi: 10.1109/ICFHR-2018.2018.00077.

[61] X. Xing *et al.*, “Potato Plant Variety Identification Study Based on Improved Swin Transformer,” *Agriculture*, vol. 15, no. 1, p. 87, Jan. 2025, doi: 10.3390/agriculture15010087.

[62] H.-Y. Li, J. A. Lawarence, P. J. Mason, and R. C. Ghail, “Fast Dynamic Time Warping and Hierarchical Clustering with Multispectral and Synthetic Aperture Radar Temporal Analysis for Unsupervised Winter Food Crop Mapping,” *Agriculture*, vol. 15, no. 1, p. 82, Jan. 2025, doi: 10.3390/agriculture15010082.

[63] J. Zhao, C. Liu, J. Han, Y. Zhou, Y. Li, and L. Zhang, “Real-Time Corn Variety Recognition Using an Efficient DenXt Architecture with Lightweight Optimizations,” *Agriculture*, vol. 15, no. 1, p. 79, Jan. 2025, doi: 10.3390/agriculture15010079.

[64] J. Han and J. H. Bae, “Developing an Hourly Water Level Prediction Model for Small- and Medium-Sized Agricultural Reservoirs Using AutoML: Case Study of Baekhak Reservoir, South Korea,” *Agriculture*, vol. 15, no. 1, p. 71, Dec. 2024, doi: 10.3390/agriculture15010071.

[65] C. Ocaña-Zuñiga, L. Quiñones-Huatangari, E. Barboza, N. C. Peña, S. H. Zamora, and J. M. P. Ojeda, “Coffee Rust Severity Analysis in Agroforestry Systems Using Deep Learning in Peruvian Tropical Ecosystems,” *Agriculture*, vol. 15, no. 1, p. 39, Dec. 2024, doi: 10.3390/agriculture15010039.

[66] M. Zhao, D. Wang, Q. Yan, Z. Li, and X. Liu, “UAV-Multispectral Based Maize Lodging Stress Assessment with Machine and Deep Learning Methods,” *Agriculture*, vol. 15, no. 1, p. 36, Dec. 2024, doi: 10.3390/agriculture15010036.

[67] Y. Song *et al.*, “Improved YOLOv8 Model for Phenotype Detection of Horticultural Seedling Growth Based on Digital Cousin,” *Agriculture*, vol. 15, no. 1, p. 28, Dec. 2024, doi: 10.3390/agriculture15010028.

[68] E. Wójcik-Gront, B. Zieniuk, and M. Pawełkowicz, “Harnessing AI-Powered Genomic Research for Sustainable Crop Improvement,” *Agriculture*, vol. 14, no. 12, p. 2299, Dec. 2024, doi: 10.3390/agriculture14122299.

[69] C. Yu, Y. Mai, C. Yang, J. Zheng, Y. Liu, and C. Yu, “IA-YOLO: A Vatica Segmentation Model Based on an Inverted Attention Block for Drone Cameras,” *Agriculture*, vol. 14, no. 12, p. 2252, Dec. 2024, doi: 10.3390/agriculture14122252.

[70] S. Zhang, H. Liu, X. Cao, and Z. Meng, “Agricultural Machinery Movement Trajectory Recognition Method Based on Two-Stage Joint Clustering,” *Agriculture*, vol. 14, no. 12, p. 2294, Dec. 2024, doi: 10.3390/agriculture14122294.

[71] D.-C. Rodríguez-Lira, D.-M. Córdova-Esparza, J. M. Álvarez-Alvarado, J. Terven, J.-A. Romero-González, and J. Rodríguez-Reséndiz, “Trends in Machine and Deep Learning Techniques for Plant Disease Identification: A Systematic Review,” *Agriculture*, vol. 14, no. 12, p. 2188, Nov. 2024, doi: 10.3390/agriculture14122188.

[72] G. Son, S. Lee, and Y. Choi, “FRESH: Fusion-Based 3D Apple Recognition via Estimating Stem Direction Heading,” *Agriculture*, vol. 14, no. 12, p. 2161, Nov. 2024, doi: 10.3390/agriculture14122161.

[73] M. Wang, W. Lai, H. Zhang, Y. Liu, and Q. Song, “Intelligent Fault Diagnosis of Inter-Turn Short Circuit Faults in PMSMs for Agricultural Machinery Based on Data Fusion and Bayesian Optimization,” *Agriculture*, vol. 14, no. 12, p. 2139, Nov. 2024, doi: 10.3390/agriculture14122139.

[74] X. Wang, Q. Wang, Y. Qiao, X. Zhang, C. Lu, and C. Wang, “Precision Weed Management for Straw-Mulched Maize Field: Advanced Weed Detection and Targeted Spraying Based on Enhanced YOLO v5s,” *Agriculture*, vol. 14, no. 12, p. 2134, Nov. 2024, doi: 10.3390/agriculture14122134.

[75] J. A. C. Martins *et al.*, “Assessment of UAV-Based Deep Learning for Corn Crop Analysis in Midwest Brazil,” *Agriculture*, vol. 14, no. 11, p. 2029, Nov. 2024, doi: 10.3390/agriculture14112029.

[76] W. Zhong, W. Yang, J. Zhu, W. Jia, X. Dong, and M. Ou, “An Improved UNet-Based Path Recognition Method in Low-Light Environments,” *Agriculture*, vol. 14, no. 11, p. 1987, Nov. 2024, doi: 10.3390/agriculture14111987.

[77] M. Liang *et al.*, “SwinLabNet: Jujube Orchard Drivable Area Segmentation Based on Lightweight CNN-Transformer Architecture,” *Agriculture*, vol. 14, no. 10, p. 1760, Oct. 2024, doi: 10.3390/agriculture14101760.

[78] T. N. Huynh, T. Burgers, and K.-D. Nguyen, “Efficient Real-Time Droplet Tracking in Crop-Spraying Systems,” *Agriculture*, vol. 14, no. 10, p. 1735, Oct. 2024, doi: 10.3390/agriculture14101735.

[79] M. Li, F. Wu, F. Wang, T. Zou, M. Li, and X. Xiao, “CNN-MLP-Based Configurable Robotic Arm for Smart Agriculture,” *Agriculture*, vol. 14, no. 9, p. 1624, Sep. 2024, doi: 10.3390/agriculture14091624.

[80] Y. Wang *et al.*, “Future Reference Evapotranspiration Trends in Shandong Province, China: Based on SAO-CNN-BiGRU-Attention and CMIP6,” *Agriculture*, vol. 14, no. 9, p. 1556, Sep. 2024, doi: 10.3390/agriculture14091556.

[81] H. Zhu, D. Wang, Y. Wei, X. Zhang, and L. Li, “Combining Transfer Learning and Ensemble Algorithms for Improved Citrus Leaf Disease Classification,” *Agriculture*, vol. 14, no. 9, p. 1549, Sep. 2024, doi: 10.3390/agriculture14091549.

[82] Y.-S. Lee, M. P. Patil, J. G. Kim, S. S. Choi, Y. B. Seo, and G.-D. Kim, “Improved Tomato Leaf Disease Recognition Based on the YOLOv5m with Various Soft Attention Module Combinations,” *Agriculture*, vol. 14, no. 9, p. 1472, Aug. 2024, doi: 10.3390/agriculture14091472.

[83] A. Ruan *et al.*, “LettuceNet: A Novel Deep Learning Approach for Efficient Lettuce Localization and Counting,” *Agriculture*, vol. 14, no. 8, p. 1412, Aug. 2024, doi: 10.3390/agriculture14081412.

[84] X. Li, M. Li, B. Liu, S. Lv, and C. Liu, “A Novel Transformer Network Based on Cross–Spatial Learning and Deformable Attention for Composite Fault Diagnosis of Agricultural Machinery Bearings,” *Agriculture*, vol. 14, no. 8, p. 1397, Aug. 2024, doi: 10.3390/agriculture14081397.

[85] S. Islam *et al.*, “Nutrient Stress Symptom Detection in Cucumber Seedlings Using Segmented Regression and a Mask Region-Based Convolutional Neural Network Model,” *Agriculture*, vol. 14, no. 8, p. 1390, Aug. 2024, doi: 10.3390/agriculture14081390.

[86] X. Zhao *et al.*, “Implementation of Large Language Models and Agricultural Knowledge Graphs for Efficient Plant Disease Detection,” *Agriculture*, vol. 14, no. 8, p. 1359, Aug. 2024, doi: 10.3390/agriculture14081359.

[87] F. Xie, E. Sun, L. Wang, G. Wang, and Q. Xiao, “Rolling Bearing Fault Diagnosis in Agricultural Machinery Based on Multi-Source Locally Adaptive Graph

Convolution," *Agriculture*, vol. 14, no. 8, p. 1333, Aug. 2024, doi: 10.3390/agriculture14081333.

[88] J. Li *et al.*, "A Multi-Area Task Path-Planning Algorithm for Agricultural Drones Based on Improved Double Deep Q-Learning Net," *Agriculture*, vol. 14, no. 8, p. 1294, Aug. 2024, doi: 10.3390/agriculture14081294.

[89] R. Maleki, F. Wu, A. Oubara, L. Fathollahi, and G. Yang, "Refinement of Cropland Data Layer with Effective Confidence Layer Interval and Image Filtering," *Agriculture*, vol. 14, no. 8, p. 1285, Aug. 2024, doi: 10.3390/agriculture14081285.

[90] M. Yuan *et al.*, "Multi-Parameter Prediction of Solar Greenhouse Environment Based on Multi-Source Data Fusion and Deep Learning," *Agriculture*, vol. 14, no. 8, p. 1245, Jul. 2024, doi: 10.3390/agriculture14081245.

[91] M. Ließ and A. Sakhaee, "Deep Learning with a Multi-Task Convolutional Neural Network to Generate a National-Scale 3D Soil Data Product: The Particle Size Distribution of the German Agricultural Soil Landscape," *Agriculture*, vol. 14, no. 8, p. 1230, Jul. 2024, doi: 10.3390/agriculture14081230.

[92] R.-F. Wang and W.-H. Su, "The Application of Deep Learning in the Whole Potato Production Chain: A Comprehensive Review," *Agriculture*, vol. 14, no. 8, p. 1225, Jul. 2024, doi: 10.3390/agriculture14081225.

[93] T. Hai *et al.*, "Implementation and Evaluation of Attention Aggregation Technique for Pear Disease Detection," *Agriculture*, vol. 14, no. 7, p. 1146, Jul. 2024, doi: 10.3390/agriculture14071146.

[94] R. Espinel, G. Herrera-Franco, J. L. Rivadeneira García, and P. Escandón-Panchana, "Artificial Intelligence in Agricultural Mapping: A Review," *Agriculture*, vol. 14, no. 7, p. 1071, Jul. 2024, doi: 10.3390/agriculture14071071.

[95] Z. Yang, W. Yang, J. Yi, and R. Liu, "WH-DETR: An Efficient Network Architecture for Wheat Spike Detection in Complex Backgrounds," *Agriculture*, vol. 14, no. 6, p. 961, Jun. 2024, doi: 10.3390/agriculture14060961.

[96] L. Qin, Z. Xu, W. Wang, and X. Wu, "YOLOv7-Based Intelligent Weed Detection and Laser Weeding System Research: Targeting Veronica didyma in Winter

Rapeseed Fields," *Agriculture*, vol. 14, no. 6, p. 910, Jun. 2024, doi: 10.3390/agriculture14060910.

[97] K. Chen *et al.*, "Integration of Image and Sensor Data for Improved Disease Detection in Peach Trees Using Deep Learning Techniques," *Agriculture*, vol. 14, no. 6, p. 797, May 2024, doi: 10.3390/agriculture14060797.

[98] J. Lu *et al.*, "Deep Learning for Multi-Source Data-Driven Crop Yield Prediction in Northeast China," *Agriculture*, vol. 14, no. 6, p. 794, May 2024, doi: 10.3390/agriculture14060794.

[99] Y. Guo *et al.*, "Online Detection of Dry Matter in Potatoes Based on Visible Near-Infrared Transmission Spectroscopy Combined with 1D-CNN," *Agriculture*, vol. 14, no. 5, p. 787, May 2024, doi: 10.3390/agriculture14050787.

[100] W. Pan, J. Wang, and W. Yang, "A Cooperative Scheduling Based on Deep Reinforcement Learning for Multi-Agricultural Machines in Emergencies," *Agriculture*, vol. 14, no. 5, p. 772, May 2024, doi: 10.3390/agriculture14050772.

[101] X. Du, D. Huang, L. Dai, and X. Du, "Recognition of Plastic Film in Terrain-Fragmented Areas Based on Drone Visible Light Images," *Agriculture*, vol. 14, no. 5, p. 736, May 2024, doi: 10.3390/agriculture14050736.

[102] H. Khalili, "Deep Learning Pricing of Processing Firms in Agricultural Markets," *Agriculture*, vol. 14, no. 5, p. 712, Apr. 2024, doi: 10.3390/agriculture14050712.

[103] C. Wang *et al.*, "Assisting the Planning of Harvesting Plans for Large Strawberry Fields through Image-Processing Method Based on Deep Learning," *Agriculture*, vol. 14, no. 4, p. 560, Apr. 2024, doi: 10.3390/agriculture14040560.

[104] C. DeChant *et al.*, "Automated Identification of Northern Leaf Blight-Infected Maize Plants from Field Imagery Using Deep Learning," *Phytopathology*, vol. 107, no. 11, pp. 1426–1432, Nov. 2017, doi: 10.1094/PHYTO-11-16-0417-R.

[105] X. Zhang, Y. Qiao, F. Meng, C. Fan, and M. Zhang, "Identification of Maize Leaf Diseases Using Improved Deep Convolutional Neural Networks," *IEEE Access*, vol. 6, pp. 30370–30377, 2018, doi: 10.1109/ACCESS.2018.2844405.

[106] J. Gao *et al.*, “Analysis of Various Machine Learning Algorithms for Using Drone Images in Livestock Farms,” *Agriculture*, vol. 14, no. 4, p. 522, Mar. 2024, doi: 10.3390/agriculture14040522.

[107] C.-H. Chang, J. Lin, J.-W. Chang, Y.-S. Huang, M.-H. Lai, and Y.-J. Chang, “Hybrid Deep Neural Networks with Multi-Tasking for Rice Yield Prediction Using Remote Sensing Data,” *Agriculture*, vol. 14, no. 4, p. 513, Mar. 2024, doi: 10.3390/agriculture14040513.

[108] M. Sibiya and M. Sumbwanyambe, “A Computational Procedure for the Recognition and Classification of Maize Leaf Diseases Out of Healthy Leaves Using Convolutional Neural Networks,” *AgriEngineering*, vol. 1, no. 1, Art. no. 1, Mar. 2019, doi: 10.3390/agriengineering1010009.

[109] A. Waheed, M. Goyal, D. Gupta, A. Khanna, A. E. Hassanien, and H. M. Pandey, “An optimized dense convolutional neural network model for disease recognition and classification in corn leaf,” *Comput. Electron. Agric.*, vol. 175, no. 105456, Jun. 2020, doi: 10.1016/j.compag.2020.105456.

[110] G. Zhang, Y. Tian, W. Yin, and C. Zheng, “An Apple Detection and Localization Method for Automated Harvesting under Adverse Light Conditions,” *Agriculture*, vol. 14, no. 3, p. 485, Mar. 2024, doi: 10.3390/agriculture14030485.

[111] D. Tamayo-Vera, X. Wang, and M. Mesbah, “A Review of Machine Learning Techniques in Agroclimatic Studies,” *Agriculture*, vol. 14, no. 3, p. 481, Mar. 2024, doi: 10.3390/agriculture14030481.

[112] L. D. Viveros Escamilla, A. Gómez-Espinosa, J. A. Escobedo Cabello, and J. A. Cantoral-Ceballos, “Maturity Recognition and Fruit Counting for Sweet Peppers in Greenhouses Using Deep Learning Neural Networks,” *Agriculture*, vol. 14, no. 3, p. 331, Feb. 2024, doi: 10.3390/agriculture14030331.

[113] R. Zhang, M. Yao, Z. Qiu, L. Zhang, W. Li, and Y. Shen, “Wheat Teacher: A One-Stage Anchor-Based Semi-Supervised Wheat Head Detector Utilizing Pseudo-Labeling and Consistency Regularization Methods,” *Agriculture*, vol. 14, no. 2, p. 327, Feb. 2024, doi: 10.3390/agriculture14020327.

[114] H. Xiong *et al.*, "DiffuCNN: Tobacco Disease Identification and Grading Model in Low-Resolution Complex Agricultural Scenes," *Agriculture*, vol. 14, no. 2, p. 318, Feb. 2024, doi: 10.3390/agriculture14020318.

[115] O. L. García-Navarrete, O. Santamaria, P. Martín-Ramos, M. Á. Valenzuela-Mahecha, and L. M. Navas-Gracia, "Development of a Detection System for Types of Weeds in Maize (Zea mays L.) under Greenhouse Conditions Using the YOLOv5 v7.0 Model," *Agriculture*, vol. 14, no. 2, p. 286, Feb. 2024, doi: 10.3390/agriculture14020286.

[116] H. Liu, K. Li, L. Ma, and Z. Meng, "Headland Identification and Ranging Method for Autonomous Agricultural Machines," *Agriculture*, vol. 14, no. 2, p. 243, Feb. 2024, doi: 10.3390/agriculture14020243.

[117] K. Dolaptsis *et al.*, "A Hybrid LSTM Approach for Irrigation Scheduling in Maize Crop," *Agriculture*, vol. 14, no. 2, p. 210, Jan. 2024, doi: 10.3390/agriculture14020210.

[118] H. Si *et al.*, "A Dual-Branch Model Integrating CNN and Swin Transformer for Efficient Apple Leaf Disease Classification," *Agriculture*, vol. 14, no. 1, p. 142, Jan. 2024, doi: 10.3390/agriculture14010142.

[119] P. Dong, K. Li, M. Wang, F. Li, W. Guo, and H. Si, "Maize Leaf Compound Disease Recognition Based on Attention Mechanism," *Agriculture*, vol. 14, no. 1, p. 74, Dec. 2023, doi: 10.3390/agriculture14010074.

[120] C.-L. Chang, H.-W. Chen, and J.-Y. Ke, "Robust Guidance and Selective Spraying Based on Deep Learning for an Advanced Four-Wheeled Farming Robot," *Agriculture*, vol. 14, no. 1, p. 57, Dec. 2023, doi: 10.3390/agriculture14010057.

[121] C. A. Aguilera, C. Figueroa-Flores, C. Aguilera, and C. Navarrete, "Comprehensive Analysis of Model Errors in Blueberry Detection and Maturity Classification: Identifying Limitations and Proposing Future Improvements in Agricultural Monitoring," *Agriculture*, vol. 14, no. 1, p. 18, Dec. 2023, doi: 10.3390/agriculture14010018.

[122] F. Grazioso, A. A. Atsapina, G. L. O. Obaeed, and N. A. Ivanova, "Deep Learning Tools for the Automatic Measurement of Coverage Area of Water-Based

Pesticide Surfactant Formulation on Plant Leaves," *Agriculture*, vol. 13, no. 12, p. 2182, Nov. 2023, doi: 10.3390/agriculture13122182.

[123] H. O. Choe and M.-H. Lee, "Artificial Intelligence-Based Fault Diagnosis and Prediction for Smart Farm Information and Communication Technology Equipment," *Agriculture*, vol. 13, no. 11, p. 2124, Nov. 2023, doi: 10.3390/agriculture13112124.

[124] Y.-J. La, D. Seo, J. Kang, M. Kim, T.-W. Yoo, and I.-S. Oh, "Deep Learning-Based Segmentation of Intertwined Fruit Trees for Agricultural Tasks," *Agriculture*, vol. 13, no. 11, p. 2097, Nov. 2023, doi: 10.3390/agriculture13112097.

[125] J. Kong, Y. Xiao, X. Jin, Y. Cai, C. Ding, and Y. Bai, "LCA-Net: A Lightweight Cross-Stage Aggregated Neural Network for Fine-Grained Recognition of Crop Pests and Diseases," *Agriculture*, vol. 13, no. 11, p. 2080, Oct. 2023, doi: 10.3390/agriculture13112080.

[126] A. Siddique, J. Sun, K. J. Hou, M. I. Vai, S. H. Pun, and M. A. Iqbal, "SpikoPoniC: A Low-Cost Spiking Neuromorphic Computer for Smart Aquaponics," *Agriculture*, vol. 13, no. 11, p. 2057, Oct. 2023, doi: 10.3390/agriculture13112057.

[127] S. C. Ibañez and C. P. Monterola, "A Global Forecasting Approach to Large-Scale Crop Production Prediction with Time Series Transformers," *Agriculture*, vol. 13, no. 9, p. 1855, Sep. 2023, doi: 10.3390/agriculture13091855.

[128] J. Kim, I. D. Ha, S. Kwon, I. Jang, and M. H. Na, "A Smart Farm DNN Survival Model Considering Tomato Farm Effect," *Agriculture*, vol. 13, no. 9, p. 1782, Sep. 2023, doi: 10.3390/agriculture13091782.

[129] Y. Sun, L. Gong, W. Zhang, B. Gao, Y. Li, and C. Liu, "Drivable Agricultural Road Region Detection Based on Pixel-Level Segmentation with Contextual Representation Augmentation," *Agriculture*, vol. 13, no. 9, p. 1736, Sep. 2023, doi: 10.3390/agriculture13091736.

[130] K. P. Panigrahi, H. Das, A. K. Sahoo, and S. C. Moharana, "Maize Leaf Disease Detection and Classification Using Machine Learning Algorithms," in *Progress in Computing, Analytics and Networking*, vol. 1119, H. Das, P. K. Pattnaik, S. S. Rautaray, and K.-C. Li, Eds., in Advances in Intelligent Systems and Computing, vol.

1119. , Singapore: Springer Singapore, 2020, pp. 659–669. doi: 10.1007/978-981-15-2414-1_66.

[131] P. Wang, E. Fan, and P. Wang, "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning," *Pattern Recognit. Lett.*, vol. 141, pp. 61–67, Jan. 2021, doi: 10.1016/j.patrec.2020.07.042.

[132] D. Xue *et al.*, "An Application of Transfer Learning and Ensemble Learning Techniques for Cervical Histopathology Image Classification," *IEEE Access*, vol. 8, pp. 104603–104618, 2020, doi: 10.1109/ACCESS.2020.2999816.

[133] N. Farmonov *et al.*, "Crop Type Classification by DESIS Hyperspectral Imagery and Machine Learning Algorithms," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 16, pp. 1576–1588, 2023, doi: 10.1109/JSTARS.2023.3239756.

[134] A. Alem and S. Kumar, "Deep Learning Models Performance Evaluations for Remote Sensed Image Classification," *IEEE Access*, vol. 10, pp. 111784–111793, 2022, doi: 10.1109/ACCESS.2022.3215264.

[135] Y. Yin and L. Wei, "Hyperspectral Image Classification Using Comprehensive Evaluation Model of Extreme Learning Machine Based on Cumulative Variation Weights," *IEEE Access*, vol. 8, pp. 187991–188003, 2020, doi: 10.1109/ACCESS.2020.3030649.

[136] S. Rui, "Medical CT Images Classification Model Based on BBO-HS Algorithm Optimized SVM," in *2020 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, Shenyang, China: IEEE, Jul. 2020, pp. 1053–1057. doi: 10.1109/ICPICS50287.2020.9202132.

[137] B. Deepa, M. Murugappan, M. G. Sumithra, M. Mahmud, and M. S. Al-Rakhami, "Pattern Descriptors Orientation and MAP Firefly Algorithm Based Brain Pathology Classification Using Hybridized Machine Learning Algorithm," *IEEE Access*, vol. 10, pp. 3848–3863, 2022, doi: 10.1109/ACCESS.2021.3100549.

[138] Z. Wang, C. Tang, X. Sima, and L. Zhang, "Research on Application of Deep Learning Algorithm in Image Classification," in *2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, Dalian, China: IEEE, Apr. 2021, pp. 1122–1125. doi: 10.1109/IPEC51340.2021.9421185.

[139] Y. Wan, T. Li, P. Wang, S. Duan, C. Zhang, and N. Li, “Robust and Efficient Classification for Underground Metal Target Using Dimensionality Reduction and Machine Learning,” *IEEE Access*, vol. 9, pp. 7384–7401, 2021, doi: 10.1109/ACCESS.2021.3049308.

[140] Md. M. Hasan *et al.*, “Enhancing Rice Crop Management: Disease Classification Using Convolutional Neural Networks and Mobile Application Integration,” *Agriculture*, vol. 13, no. 8, p. 1549, Aug. 2023, doi: 10.3390/agriculture13081549.

[141] D. Radočaj, I. Plaščak, and M. Jurišić, “Global Navigation Satellite Systems as State-of-the-Art Solutions in Precision Agriculture: A Review of Studies Indexed in the Web of Science,” *Agriculture*, vol. 13, no. 7, p. 1417, Jul. 2023, doi: 10.3390/agriculture13071417.

[142] Q. Yang, Y. Ye, L. Gu, and Y. Wu, “MSFCA-Net: A Multi-Scale Feature Convolutional Attention Network for Segmenting Crops and Weeds in the Field,” *Agriculture*, vol. 13, no. 6, p. 1176, May 2023, doi: 10.3390/agriculture13061176.

[143] F. Huang *et al.*, “Interpreting Conv-LSTM for Spatio-Temporal Soil Moisture Prediction in China,” *Agriculture*, vol. 13, no. 5, p. 971, Apr. 2023, doi: 10.3390/agriculture13050971.

[144] D. Čirjak, I. Aleksi, D. Lemic, and I. Pajač Živković, “EfficientDet-4 Deep Neural Network-Based Remote Monitoring of Codling Moth Population for Early Damage Detection in Apple Orchard,” *Agriculture*, vol. 13, no. 5, p. 961, Apr. 2023, doi: 10.3390/agriculture13050961.

[145] M. Aggarwal *et al.*, “Pre-Trained Deep Neural Network-Based Features Selection Supported Machine Learning for Rice Leaf Disease Classification,” *Agriculture*, vol. 13, no. 5, p. 936, Apr. 2023, doi: 10.3390/agriculture13050936.

[146] X. Jin, C. Lin, J. Ji, W. Li, B. Zhang, and H. Suo, “An Inter-Ridge Navigation Path Extraction Method Based on Res2net50 Segmentation Model,” *Agriculture*, vol. 13, no. 4, p. 881, Apr. 2023, doi: 10.3390/agriculture13040881.

[147] T. Nazir, M. M. Iqbal, S. Jabbar, A. Hussain, and M. Albathan, “EfficientPNet—An Optimized and Efficient Deep Learning Approach for Classifying Disease of

Potato Plant Leaves," *Agriculture*, vol. 13, no. 4, p. 841, Apr. 2023, doi: 10.3390/agriculture13040841.

[148] Z. Ullah, N. Alsubaie, M. Jamjoom, S. H. Alajmani, and F. Saleem, "EffiMob-Net: A Deep Learning-Based Hybrid Model for Detection and Identification of Tomato Diseases Using Leaf Images," *Agriculture*, vol. 13, no. 3, p. 737, Mar. 2023, doi: 10.3390/agriculture13030737.

[149] A. C. Teixeira, J. Ribeiro, R. Morais, J. J. Sousa, and A. Cunha, "A Systematic Review on Automatic Insect Detection Using Deep Learning," *Agriculture*, vol. 13, no. 3, p. 713, Mar. 2023, doi: 10.3390/agriculture13030713.

[150] S. Lin, Y. Xiu, J. Kong, C. Yang, and C. Zhao, "An Effective Pyramid Neural Network Based on Graph-Related Attentions Structure for Fine-Grained Disease and Pest Identification in Intelligent Agriculture," *Agriculture*, vol. 13, no. 3, p. 567, Feb. 2023, doi: 10.3390/agriculture13030567.

[151] M. Albahar, "A Survey on Deep Learning and Its Impact on Agriculture: Challenges and Opportunities," *Agriculture*, vol. 13, no. 3, p. 540, Feb. 2023, doi: 10.3390/agriculture13030540.

[152] Y. Li, H. Liu, J. Wei, X. Ma, G. Zheng, and L. Xi, "Research on Winter Wheat Growth Stages Recognition Based on Mobile Edge Computing," *Agriculture*, vol. 13, no. 3, p. 534, Feb. 2023, doi: 10.3390/agriculture13030534.

[153] M. Khalid, M. Sarfraz, U. Iqbal, M. Aftab, G. Niedbała, and H. Rauf, "Real-Time Plant Health Detection Using Deep Convolutional Neural Networks," *Agriculture*, vol. 13, no. 2, p. 510, Feb. 2023, doi: 10.3390/agriculture13020510.

[154] G. J. Q. Vasconcelos, G. S. R. Costa, T. V. Spina, and H. Pedrini, "Low-Cost Robot for Agricultural Image Data Acquisition," *Agriculture*, vol. 13, no. 2, p. 413, Feb. 2023, doi: 10.3390/agriculture13020413.

[155] C. Badgujar, S. Das, D. M. Figueroa, and D. Flippo, "Application of Computational Intelligence Methods in Agricultural Soil–Machine Interaction: A Review," *Agriculture*, vol. 13, no. 2, p. 357, Jan. 2023, doi: 10.3390/agriculture13020357.

[156] G. G. R. D. Castro *et al.*, “Adaptive Path Planning for Fusing Rapidly Exploring Random Trees and Deep Reinforcement Learning in an Agriculture Dynamic Environment UAVs,” *Agriculture*, vol. 13, no. 2, p. 354, Jan. 2023, doi: 10.3390/agriculture13020354.

[157] H. Orchi, M. Sadik, M. Khaldoun, and E. Sabir, “Automation of Crop Disease Detection through Conventional Machine Learning and Deep Transfer Learning Approaches,” *Agriculture*, vol. 13, no. 2, p. 352, Jan. 2023, doi: 10.3390/agriculture13020352.

[158] C. E. Hachimi, S. Belaqziz, S. Khabba, B. Sebbar, D. Dhiba, and A. Chehbouni, “Smart Weather Data Management Based on Artificial Intelligence and Big Data Analytics for Precision Agriculture,” *Agriculture*, vol. 13, no. 1, p. 95, Dec. 2022, doi: 10.3390/agriculture13010095.

[159] D. Čirjak *et al.*, “Monitoring System for Leucoptera malifoliella (O. Costa, 1836) and Its Damage Based on Artificial Neural Networks,” *Agriculture*, vol. 13, no. 1, p. 67, Dec. 2022, doi: 10.3390/agriculture13010067.

[160] R. Ma *et al.*, “Identification of Maize Seed Varieties Using MobileNetV2 with Improved Attention Mechanism CBAM,” *Agriculture*, vol. 13, no. 1, p. 11, Dec. 2022, doi: 10.3390/agriculture13010011.

[161] Z. Yao, X. Zhu, Y. Zeng, and X. Qiu, “Extracting Tea Plantations from Multitemporal Sentinel-2 Images Based on Deep Learning Networks,” *Agriculture*, vol. 13, no. 1, p. 10, Dec. 2022, doi: 10.3390/agriculture13010010.

[162] D. Pang, H. Wang, P. Chen, and D. Liang, “Spider Mites Detection in Wheat Field Based on an Improved RetinaNet,” *Agriculture*, vol. 12, no. 12, p. 2160, Dec. 2022, doi: 10.3390/agriculture12122160.

[163] Y. Li, B. Zhao, W. Zhang, L. Wei, and L. Zhou, “Evaluation of Agricultural Machinery Operational Benefits Based on Semi-Supervised Learning,” *Agriculture*, vol. 12, no. 12, p. 2075, Dec. 2022, doi: 10.3390/agriculture12122075.

[164] L. Liu, Q. Bi, J. Liang, Z. Li, W. Wang, and Q. Zheng, “Farmland Soil Block Identification and Distribution Statistics Based on Deep Learning,” *Agriculture*, vol. 12, no. 12, p. 2038, Nov. 2022, doi: 10.3390/agriculture12122038.

[165] D. Wang, Z. Li, X. Du, Z. Ma, and X. Liu, “Farmland Obstacle Detection from the Perspective of UAVs Based on Non-local Deformable DETR,” *Agriculture*, vol. 12, no. 12, p. 1983, Nov. 2022, doi: 10.3390/agriculture12121983.

[166] Y. Huang, P. Xia, L. Gong, B. Chen, Y. Li, and C. Liu, “Designing an Interactively Cognitive Humanoid Field-Phenotyping Robot for In-Field Rice Tiller Counting,” *Agriculture*, vol. 12, no. 11, p. 1966, Nov. 2022, doi: 10.3390/agriculture12111966.

[167] J. Suto, “A Novel Plug-in Board for Remote Insect Monitoring,” *Agriculture*, vol. 12, no. 11, p. 1897, Nov. 2022, doi: 10.3390/agriculture12111897.

[168] R. C. Bernardes *et al.*, “Deep-Learning Approach for Fusarium Head Blight Detection in Wheat Seeds Using Low-Cost Imaging Technology,” *Agriculture*, vol. 12, no. 11, p. 1801, Oct. 2022, doi: 10.3390/agriculture12111801.

[169] M. Fraiwan, E. Faouri, and N. Khasawneh, “Multiclass Classification of Grape Diseases Using Deep Artificial Intelligence,” *Agriculture*, vol. 12, no. 10, p. 1542, Sep. 2022, doi: 10.3390/agriculture12101542.

[170] A. I. Jajja *et al.*, “Compact Convolutional Transformer (CCT)-Based Approach for Whitefly Attack Detection in Cotton Crops,” *Agriculture*, vol. 12, no. 10, p. 1529, Sep. 2022, doi: 10.3390/agriculture12101529.

[171] B. Tugrul, E. Elfatimi, and R. Eryigit, “Convolutional Neural Networks in Detection of Plant Leaf Diseases: A Review,” *Agriculture*, vol. 12, no. 8, p. 1192, Aug. 2022, doi: 10.3390/agriculture12081192.

[172] X. Li, J. Du, J. Yang, and S. Li, “When Mobilenetv2 Meets Transformer: A Balanced Sheep Face Recognition Model,” *Agriculture*, vol. 12, no. 8, p. 1126, Jul. 2022, doi: 10.3390/agriculture12081126.

[173] T. Kim, H. Kim, K. Baik, and Y. Choi, “Instance-Aware Plant Disease Detection by Utilizing Saliency Map and Self-Supervised Pre-Training,” *Agriculture*, vol. 12, no. 8, p. 1084, Jul. 2022, doi: 10.3390/agriculture12081084.

[174] W. Li, T. Zhu, X. Li, J. Dong, and J. Liu, “Recommending Advanced Deep Learning Models for Efficient Insect Pest Detection,” *Agriculture*, vol. 12, no. 7, p. 1065, Jul. 2022, doi: 10.3390/agriculture12071065.

[175] N. Mamat, M. F. Othman, R. Abdoulghafor, S. B. Belhaouari, N. Mamat, and S. F. Mohd Hussein, "Advanced Technology in Agriculture Industry by Implementing Image Annotation Technique and Deep Learning Approach: A Review," *Agriculture*, vol. 12, no. 7, p. 1033, Jul. 2022, doi: 10.3390/agriculture12071033.

[176] J. Escorcia-Gutierrez, M. Gamarra, R. Soto-Diaz, M. Pérez, N. Madera, and R. F. Mansour, "Intelligent Agricultural Modelling of Soil Nutrients and pH Classification Using Ensemble Deep Learning Techniques," *Agriculture*, vol. 12, no. 7, p. 977, Jul. 2022, doi: 10.3390/agriculture12070977.

[177] H. Waheed, N. Zafar, W. Akram, A. Manzoor, A. Gani, and S. U. Islam, "Deep Learning Based Disease, Pest Pattern and Nutritional Deficiency Detection System for 'Zingiberaceae' Crop," *Agriculture*, vol. 12, no. 6, p. 742, May 2022, doi: 10.3390/agriculture12060742.

[178] Y. Hwang, S. Lee, T. Kim, K. Baik, and Y. Choi, "Crop Growth Monitoring System in Vertical Farms Based on Region-of-Interest Prediction," *Agriculture*, vol. 12, no. 5, p. 656, Apr. 2022, doi: 10.3390/agriculture12050656.

[179] D.-C. Hsiou, F. Huang, F. J. Tey, T.-Y. Wu, and Y.-C. Lee, "An Automated Crop Growth Detection Method Using Satellite Imagery Data," *Agriculture*, vol. 12, no. 4, p. 504, Apr. 2022, doi: 10.3390/agriculture12040504.

[180] Y. H. Gu, H. Yin, D. Jin, R. Zheng, and S. J. Yoo, "Improved Multi-Plant Disease Recognition Method Using Deep Convolutional Neural Networks in Six Diseases of Apples and Pears," *Agriculture*, vol. 12, no. 2, p. 300, Feb. 2022, doi: 10.3390/agriculture12020300.

[181] M. A. Amani and F. Marinello, "A Deep Learning-Based Model to Reduce Costs and Increase Productivity in the Case of Small Datasets: A Case Study in Cotton Cultivation," *Agriculture*, vol. 12, no. 2, p. 267, Feb. 2022, doi: 10.3390/agriculture12020267.

[182] L. Du *et al.*, "A Novel Object Detection Model Based on Faster R-CNN for Spodoptera frugiperda According to Feeding Trace of Corn Leaves," *Agriculture*, vol. 12, no. 2, p. 248, Feb. 2022, doi: 10.3390/agriculture12020248.

[183] P. Gao *et al.*, “Modeling for the Prediction of Soil Moisture in Litchi Orchard with Deep Long Short-Term Memory,” *Agriculture*, vol. 12, no. 1, p. 25, Dec. 2021, doi: 10.3390/agriculture12010025.

[184] H. Orchi, M. Sadik, and M. Khaldoun, “On Using Artificial Intelligence and the Internet of Things for Crop Disease Detection: A Contemporary Survey,” *Agriculture*, vol. 12, no. 1, p. 9, Dec. 2021, doi: 10.3390/agriculture12010009.

[185] C.-L. Chang, B.-X. Xie, and S.-C. Chung, “Mechanical Control with a Deep Learning Method for Precise Weeding on a Farm,” 2021.

[186] Y. Peng, A. Wang, J. Liu, and M. Faheem, “A Comparative Study of Semantic Segmentation Models for Identification of Grape with Different Varieties,” *Agriculture*, vol. 11, no. 10, p. 997, Oct. 2021, doi: 10.3390/agriculture11100997.

[187] X. Guo *et al.*, “ACE-ADP: Adversarial Contextual Embeddings Based Named Entity Recognition for Agricultural Diseases and Pests,” *Agriculture*, vol. 11, no. 10, p. 912, Sep. 2021, doi: 10.3390/agriculture11100912.

[188] X.-B. Jin *et al.*, “Deep-Learning Temporal Predictor via Bidirectional Self-Attentive Encoder–Decoder Framework for IOT-Based Environmental Sensing in Intelligent Greenhouse,” *Agriculture*, vol. 11, no. 8, p. 802, Aug. 2021, doi: 10.3390/agriculture11080802.

[189] S. Zhao, Y. Peng, J. Liu, and S. Wu, “Tomato Leaf Disease Diagnosis Based on Improved Convolution Neural Network by Attention Module,” 2021.

[190] S. Nosratabadi, S. Ardabili, Z. Lakner, C. Mako, and A. Mosavi, “Prediction of Food Production Using Machine Learning Algorithms of Multilayer Perceptron and ANFIS,” *Agriculture*, vol. 11, no. 5, p. 408, May 2021, doi: 10.3390/agriculture11050408.

[191] E. Pagliarino and S. Rolfo, “Examining Researchers’ Attitudes, Barriers, and Opportunities for Participatory Research: The Case of the Riso-Biosystems Project on Organic Rice,” *Agriculture*, vol. 11, no. 5, p. 376, Apr. 2021, doi: 10.3390/agriculture11050376.

[192] A. S. Aguiar *et al.*, “Bringing Semantics to the Vineyard: An Approach on Deep Learning-Based Vine Trunk Detection,” *Agriculture*, vol. 11, no. 2, p. 131, Feb. 2021, doi: 10.3390/agriculture11020131.

[193] V. S. Yakkala, K. V. Nusimala, B. Gayathri, S. S. Aravinth, A. O. Salau, and S. Srithar, “Deep learning-based crop health enhancement through early disease prediction”.

[194] A. Oikonomidis, C. Catal, and A. Kassahun, “Deep learning for crop yield prediction: a systematic literature review,” *N. Z. J. Crop Hortic. Sci.*, vol. 51, no. 1, pp. 1–26, Jan. 2023, doi: 10.1080/01140671.2022.2032213.

[195] S. Zhang, R. Jing, and X. Shi, “Crop pest recognition based on a modified capsule network,” *Syst. Sci. Control Eng.*, vol. 10, no. 1, pp. 552–561, Dec. 2022, doi: 10.1080/21642583.2022.2074168.

[196] G. Singh and K. K. Yogi, “Performance evaluation of plant leaf disease detection using deep learning models,” *Arch. Phytopathol. Plant Prot.*, vol. 56, no. 3, pp. 209–233, Feb. 2023, doi: 10.1080/03235408.2023.2183792.

[197] K. Rangarajan Aravind and P. Raja, “Automated disease classification in (Selected) agricultural crops using transfer learning,” *Automatika*, vol. 61, no. 2, pp. 260–272, Apr. 2020, doi: 10.1080/00051144.2020.1728911.

[198] M. Andargie, M. Abera, A. Tesfaye, and E. Demis, “Occurrence, distribution, and management experiences of rice (*Oryza sativa* L.) major diseases and pests in Ethiopia: a review,” *Cogent Food Agric.*, vol. 10, no. 1, Dec. 2024, doi: 10.1080/23311932.2023.2300558.

[199] J. S.S. and S. G. S., “Artificial intelligence approach for diagnosing banana crop diseases through image processing framework: an insightful survey journey from farmers field to digital cloud,” *Arch. Phytopathol. Plant Prot.*, vol. 57, no. 3, pp. 163–199, Feb. 2024, doi: 10.1080/03235408.2024.2347966.

[200] M. Sharma, C. J. Kumar, and A. Deka, “Early diagnosis of rice plant disease using machine learning techniques,” *Arch. Phytopathol. Plant Prot.*, vol. 55, no. 3, pp. 259–283, Feb. 2022, doi: 10.1080/03235408.2021.2015866.

[201] G. S. Mmbando, “Harnessing artificial intelligence and remote sensing in climate-smart agriculture: the current strategies needed for enhancing global food security,” *Cogent Food Agric.*, vol. 11, no. 1, Dec. 2025, doi: 10.1080/23311932.2025.2454354.

[202] M. Sharma, C. J. Kumar, T. P. Singh, J. Talukdar, R. K. Sharma, and A. Ganguly, “Enhancing disease region segmentation in rice leaves using modified deep learning architectures,” *Arch. Phytopathol. Plant Prot.*, vol. 56, no. 20, pp. 1555–1580, Dec. 2023, doi: 10.1080/03235408.2024.2310326.

[203] J. Li, Y. Cai, Q. Li, M. Kou, and T. Zhang, “A review of remote sensing image segmentation by deep learning methods,” *Int. J. Digit. Earth*, vol. 17, no. 1, Dec. 2024, doi: 10.1080/17538947.2024.2328827.

[204] R. Deshpande and H. Patidar, “Tomato plant leaf disease detection using generative adversarial network and deep convolutional neural network,” *Imaging Sci. J.*, vol. 70, no. 1, pp. 1–9, Jan. 2022, doi: 10.1080/13682199.2022.2161696.

[205] M. Nath, P. Mitra, and D. Kumar, “A novel residual learning-based deep learning model integrated with attention mechanism and SVM for identifying tea plant diseases,” *Int. J. Comput. Appl.*, vol. 45, no. 6, pp. 471–484, Jun. 2023, doi: 10.1080/1206212X.2023.2235750.

[206] Y. Hang, X. Meng, R. Wang, X. Tian, and J. Cui, “Ensemble Technique of Deep Learning Model for Identifying Tomato Leaf Diseases Based on Choquet Fuzzy Integral,” *Cybern. Syst.*, pp. 1–24, May 2024, doi: 10.1080/01969722.2024.2343993.

[207] L. Goel and J. Nagpal, “A Systematic Review of Recent Machine Learning Techniques for Plant Disease Identification and Classification,” *IETE Tech. Rev.*, vol. 40, no. 3, pp. 423–439, May 2023, doi: 10.1080/02564602.2022.2121772.

[208] S. Nain, N. Mittal, and M. Hanmandlu, “CNN-based plant disease recognition using colour space models,” *Int. J. Image Data Fusion*, vol. 15, no. 3, pp. 373–386, Jul. 2024, doi: 10.1080/19479832.2023.2300335.

[209] N. Bharathi Raja and P. Selvi Rajendran, “An efficient banana plant leaf disease classification using optimal ensemble deep transfer network,” *J. Exp. Theor. Artif. Intell.*, pp. 1–24, Aug. 2023, doi: 10.1080/0952813X.2023.2241867.

[210] R. Deshpande and H. Patidar, “Detection of leaf disease in tomato plants using a lightweight parallel deep convolutional neural network,” *Arch. Phytopathol. Plant Prot.*, vol. 56, no. 9, pp. 707–720, May 2023, doi: 10.1080/03235408.2023.2216359.

[211] H. J. Escalante, S. Rodríguez-Sánchez, M. Jiménez-Lizárraga, A. Morales-Reyes, J. De La Calleja, and R. Vazquez, “Barley yield and fertilization analysis from UAV imagery: a deep learning approach,” *Int. J. Remote Sens.*, vol. 40, no. 7, pp. 2493–2516, Apr. 2019, doi: 10.1080/01431161.2019.1577571.

[212] S. P. K. Surata, I. B. A. Arjaya, I. G. A. S. R. Jayantini, and I. G. A. P. E. Putri, “Exploring biosecurity education through photovoice: student teachers’ perspectives on pests and diseases,” *Res. Sci. Technol. Educ.*, pp. 1–19, Oct. 2024, doi: 10.1080/02635143.2024.2412027.

[213] S. Verma, P. Kumar, and J. P. Singh, “A Unified Lightweight CNN-based Model for Disease Detection and Identification in Corn, Rice, and Wheat,” *IETE J. Res.*, vol. 70, no. 3, pp. 2481–2492, Mar. 2024, doi: 10.1080/03772063.2023.2181229.

[214] T. Y. Mahesh and M. P. Mathew, “Detection of Bacterial Spot Disease in Bell Pepper Plant Using YOLOv3,” *IETE J. Res.*, vol. 70, no. 3, pp. 2583–2590, Mar. 2024, doi: 10.1080/03772063.2023.2176367.

[215] R. Ansari, A. Manna, S. Hazra, S. Bose, A. Chatterjee, and P. Sen, “Breeding 4.0 vis-à-vis application of artificial intelligence (AI) in crop improvement: an overview,” *N. Z. J. Crop Hortic. Sci.*, vol. 53, no. 4, pp. 812–854, Oct. 2025, doi: 10.1080/01140671.2024.2409775.

[216] F. Lin, Y. Ji, and S. Xu, “Sweetener identification using transfer learning and attention mechanism,” *CyTA - J. Food*, vol. 22, no. 1, Dec. 2024, doi: 10.1080/19476337.2024.2341812.

[217] L. Zhao, M. W. Harding, G. Peng, R. Lange, S. Walkowiak, and W. G. D. Fernando, “Artificial intelligence analysis of contributive factors in determining blackleg disease severity in”.

[218] A. Sreedevi and C. Manike, “Development of weighted ensemble transfer learning for tomato leaf disease classification solving low resolution problems,”

Imaging Sci. J., vol. 71, no. 2, pp. 161–187, Feb. 2023, doi: 10.1080/13682199.2023.2178605.

[219] M. M, M. Akila, and P. B, "A review of plant disease detection and classification methods," *Int. J. Model. Simul.*, vol. 45, no. 3, pp. 1074–1090, May 2025, doi: 10.1080/02286203.2023.2254192.

[220] M. El Sakka, J. Mothe, and M. Ivanovici, "Images and CNN applications in smart agriculture," *Eur. J. Remote Sens.*, vol. 57, no. 1, p. 2352386, Dec. 2024, doi: 10.1080/22797254.2024.2352386.

[221] G. Hu *et al.*, "Detection of diseased pine trees in unmanned aerial vehicle images by using deep convolutional neural networks," *Geocarto Int.*, vol. 37, no. 12, pp. 3520–3539, Jun. 2022, doi: 10.1080/10106049.2020.1864025.

[222] Z. Wei, Y. Jiang, and H. Liu, "Optical imaging combined with artificial intelligence in plant disease detection: a comprehensive review," *Spectrosc. Lett.*, pp. 1–25, Feb. 2025, doi: 10.1080/00387010.2025.2459246.

[223] M. Urbieta, M. Urbieta, M. Pereyra, T. Laborde, G. Villarreal, and M. Del Pino, "A scalable offline AI-based solution to assist the diseases and plague detection in agriculture," *J. Decis. Syst.*, vol. 33, no. 3, pp. 459–476, Jul. 2024, doi: 10.1080/12460125.2023.2226381.

[224] S. Pandey, R. Miri, G. R. Sinha, and R. Raja, "AFD filter and E^2N^2 classifier for improving visualization of crop image and crop classification in remote sensing image," *Int. J. Remote Sens.*, vol. 43, no. 15–16, pp. 5848–5873, Aug. 2022, doi: 10.1080/01431161.2021.2000062.

[225] P. Pathmanaban, B. K. Gnanavel, and S. S. Anandan, "Guava fruit (*Psidium guajava*) damage and disease detection using deep convolutional neural networks and thermal imaging," *Imaging Sci. J.*, vol. 70, no. 2, pp. 102–116, Feb. 2022, doi: 10.1080/13682199.2022.2163536.

[226] D. J. Chaudhari and M. K., "FRRSA: Fractional Remora Reptile Search Algorithm-based LeNet for rice leaf disease classification," *Aust. J. Electr. Electron. Eng.*, pp. 1–19, Nov. 2024, doi: 10.1080/1448837x.2024.2413226.

[227] G. Taş and C. Bal, “Bloomed or non-bloomed fruit tree classification with transfer learning,” *Remote Sens. Lett.*, vol. 14, no. 9, pp. 981–992, Sep. 2023, doi: 10.1080/2150704x.2023.2258457.

[228] N. Ahmad, S. Qadri, and N. Akhtar, “A multimodal fusion framework to diagnose cotton leaf curl virus using machine vision techniques,” *Cogent Food Agric.*, vol. 10, no. 1, Dec. 2024, doi: 10.1080/23311932.2024.2339572.

[229] K. N. Vhatkar, V. V. Kimbahune, R. K. Somkunwar, R. Bharti, J. M. Sarwade, and A. B. Kathole, “An intelligent framework of groundnut plant disease classification using heuristic approach-aided multilevel K-means clustering and attention-based hybrid dilated residual network,” *Aust. J. Electr. Electron. Eng.*, pp. 1–30, Jul. 2025, doi: 10.1080/1448837X.2025.2529098.

[230] A. Sreedevi and C. Manike, “A Smart Solution for Tomato Leaf Disease Classification by Modified Recurrent Neural Network with Severity Computation,” *Cybern. Syst.*, vol. 55, no. 2, pp. 409–449, Feb. 2024, doi: 10.1080/01969722.2022.2122004.

[231] A. Kshetrimayum, A. Goyal, R. H, and B. K. Bhadra, “Semi physical and machine learning approach for yield estimation of pearl millet crop using SAR and optical data products,” *J. Spat. Sci.*, vol. 69, no. 2, pp. 573–592, Apr. 2024, doi: 10.1080/14498596.2023.2259857.

[232] M. Rehman *et al.*, “Semantics Analysis of Agricultural Experts’ Opinions for Crop Productivity through Machine Learning,” *Appl. Artif. Intell.*, vol. 36, no. 1, p. 2012055, Dec. 2022, doi: 10.1080/08839514.2021.2012055.

[233] R. Singh, P. Krishnan, V. K. Singh, and B. Das, “Estimation of yellow rust severity in wheat using visible and thermal imaging coupled with machine learning models,” *Geocarto Int.*, vol. 38, no. 1, p. 2160831, Dec. 2023, doi: 10.1080/10106049.2022.2160831.

[234] C.-S. Huang and A. Sayed, “Novel forecasting of white maize futures volatility: a hybrid GARCH-based bi-directional LSTM model,” *Cogent Econ. Finance*, vol. 13, no. 1, p. 2484422, Dec. 2025, doi: 10.1080/23322039.2025.2484422.

[235] A. Kaur, Geetanjali, M. Singh, A. Mittal, R. Mittal, and V. Malik, “Modified HDBSCAN based segmentation hyperspectral image segmentation for cotton crop classification,” *Int. J. Image Data Fusion*, vol. 15, no. 2, pp. 232–256, Apr. 2024, doi: 10.1080/19479832.2024.2321887.

[236] B. Swaminathan, S. Palani, and S. Vairavasundaram, “Meta Learning-Based Dynamic Ensemble Model for Crop Selection,” *Appl. Artif. Intell.*, vol. 36, no. 1, p. 2145646, Dec. 2022, doi: 10.1080/08839514.2022.2145646.

[237] K. P. Chaitanya and A. M. Posonia, “Tuned weighted feature fusion with hybridized DNN-RNN framework for plant disease detection and classification,” *Int. J. Remote Sens.*, vol. 45, no. 5, pp. 1608–1639, Mar. 2024, doi: 10.1080/01431161.2024.2313995.

[238] Y. Bahaddou, L. Tamym, and L. Benyoucef, “Deep learning for freshness categorisation in sustainable agricultural supply chains: a focus on quality assessment of fruits and vegetables,” *Supply Chain Forum Int. J.*, pp. 1–24, Nov. 2024, doi: 10.1080/16258312.2024.2428155.

[239] B. Wang, “Identification of Crop Diseases and Insect Pests Based on Deep Learning,” *Sci. Program.*, vol. 2022, pp. 1–10, Jan. 2022, doi: 10.1155/2022/9179998.

[240] Y. Yu, “Research Progress of Crop Disease Image Recognition Based on Wireless Network Communication and Deep Learning,” *Wirel. Commun. Mob. Comput.*, vol. 2021, pp. 1–15, Oct. 2021, doi: 10.1155/2021/7577349.

[241] N. Biradar and G. Hosalli, “Segmentation and detection of crop pests using novel U-NET with hybrid deep learning mechanism,” *Pest Manag. Sci.*, vol. 80, no. 8, pp. 3795–3807, Aug. 2024, doi: 10.1002/ps.8083.

[242] D. Liu, H. Yang, Y. Gong, and Q. Chen, “A Recognition Method of Crop Diseases and Insect Pests Based on Transfer Learning and Convolution Neural Network,” *Math. Probl. Eng.*, vol. 2022, pp. 1–10, Jul. 2022, doi: 10.1155/2022/1470541.

[243] G. Ganesan and J. Chinnappan, “Hybridization of ResNet with YOLO classifier for automated paddy leaf disease recognition: An optimized model,” *J. Field Robot.*, vol. 39, no. 7, pp. 1085–1109, Oct. 2022, doi: 10.1002/rob.22089.

[244] G. K S, S. K. P, R. A, and G. K, “Crop pest identification using deep network based extracted features and MobileENet in smart agriculture,” *Land Degrad. Dev.*, vol. 35, no. 11, pp. 3642–3652, Jul. 2024, doi: 10.1002/ldr.5157.

[245] Y. Liu, X. Zhang, Y. Gao, T. Qu, and Y. Shi, “Improved CNN Method for Crop Pest Identification Based on Transfer Learning,” *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–8, Mar. 2022, doi: 10.1155/2022/9709648.

[246] Y. Lin, J. Xu, and Y. Zhang, “Identification Method of Citrus Aurantium Diseases and Pests Based on Deep Convolutional Neural Network,” *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–8, May 2022, doi: 10.1155/2022/7012399.

[247] N. N. Ahmad Loti, M. R. Mohd Noor, and S. Chang, “Integrated analysis of machine learning and deep learning in chili pest and disease identification,” *J. Sci. Food Agric.*, vol. 101, no. 9, pp. 3582–3594, Jul. 2021, doi: 10.1002/jsfa.10987.

[248] T. Daniya and V. Srinivasan, “Shuffled shepherd social optimization based deep learning for rice leaf disease classification and severity percentage prediction,” *Concurr. Comput. Pract. Exp.*, vol. 35, no. 4, p. e7523, Feb. 2023, doi: 10.1002/cpe.7523.

[249] Y. He, H. Zeng, Y. Fan, S. Ji, and J. Wu, “Application of Deep Learning in Integrated Pest Management: A Real-Time System for Detection and Diagnosis of Oilseed Rape Pests,” *Mob. Inf. Syst.*, vol. 2019, pp. 1–14, Jul. 2019, doi: 10.1155/2019/4570808.

[250] S. K. Upadhyay and A. K. Dwivedi, “Rice Plant Disease Diagnosis Using SQUEEZENET and Deep Transfer Learning,” *J. Phytopathol.*, vol. 173, no. 3, p. e70092, May 2025, doi: 10.1111/jph.70092.

[251] E. Miao, G. Zhou, and S. Zhao, “Research on Soybean Disease Identification Method Based on Deep Learning,” *Mob. Inf. Syst.*, vol. 2022, pp. 1–8, Aug. 2022, doi: 10.1155/2022/1952936.

[252] M. Shereesha, G. K. Sandhia, and R. Pitchai, “Dual RNN Architecture for Crop Disease Detection: Improved Patch-Based CNN for Segmentation,” *J. Phytopathol.*, vol. 173, no. 3, p. e70079, May 2025, doi: 10.1111/jph.70079.

[253] P. P. Nimbhore, R. Tiwari, T. Hazra, and M. P. Yadav, “Classification of cotton crop disease using hybrid model and MDFC feature extraction method,” *J. Phytopathol.*, vol. 172, no. 4, p. e13324, Jul. 2024, doi: 10.1111/jph.13324.

[254] G. Polder, P. M. Blok, T. Van Daalen, J. Peller, and N. Mylonas, “A Smart Camera With Integrated Deep Learning Processing for Disease Detection in Open Field Crops of Grape, Apple, and Carrot,” *J. Field Robot.*, vol. 42, no. 5, pp. 2062–2075, Aug. 2025, doi: 10.1002/rob.22510.

[255] R. N. Atole, N. B Pokale, and A. D. Patil, “Crop Leaf Segmentation and Disease Detection Based on Shepherd Wide Residual Network,” *J. Phytopathol.*, vol. 173, no. 3, p. e70080, May 2025, doi: 10.1111/jph.70080.

[256] M. Dong *et al.*, “Measurement Method of Plant Phenotypic Parameters Based on Image Deep Learning,” *Wirel. Commun. Mob. Comput.*, vol. 2022, pp. 1–9, Apr. 2022, doi: 10.1155/2022/7664045.

[257] Y. He, Q. Gao, and Z. Ma, “A Crop Leaf Disease Image Recognition Method Based on Bilinear Residual Networks,” *Math. Probl. Eng.*, vol. 2022, pp. 1–15, Apr. 2022, doi: 10.1155/2022/2948506.

[258] J. Wang, Y. Chen, J. Huang, X. Jiang, and K. Wan, “Leveraging machine learning for advancing insect pest control: A bibliometric analysis,” *J. Appl. Entomol.*, vol. 149, no. 3, pp. 293–308, Apr. 2025, doi: 10.1111/jen.13223.

[259] Q. Tian *et al.*, “Enhancing practicality of deep learning for crop disease identification under field conditions: insights from model evaluation and crop-specific approaches,” *Pest Manag. Sci.*, vol. 80, no. 11, pp. 5864–5875, Nov. 2024, doi: 10.1002/ps.8317.

[260] A. Emmanuel, R. W. Mwangi, P. Murithi, K. Fredrick, and T. Danison, “Classification of cassava leaf diseases using deep Gaussian transfer learning model,” *Eng. Rep.*, vol. 5, no. 9, p. e12651, Sep. 2023, doi: 10.1002/eng2.12651.

[261] G. Kaur, F. A. M. Al-Yarimi, S. Bharany, A. U. Rehman, and S. Hussen, “Explainable AI for Cotton Leaf Disease Classification: A Metaheuristic-Optimized Deep Learning Approach,” *Food Sci. Nutr.*, vol. 13, no. 7, p. e70658, Jul. 2025, doi: 10.1002/fsn3.70658.

[262] M. Mittal, V. Gupta, M. Aamash, and T. Upadhyay, “Machine learning for pest detection and infestation prediction: A comprehensive review,” *WIREs Data Min. Knowl. Discov.*, vol. 14, no. 5, p. e1551, Sep. 2024, doi: 10.1002/widm.1551.

[263] I. Ahmed and P. K. Yadav, “Plant disease detection using machine learning approaches,” *Expert Syst.*, vol. 40, no. 5, p. e13136, Jun. 2023, doi: 10.1111/exsy.13136.

[264] J. Arun Pandian and K. Kanchanadevi, “An improved deep convolutional neural network for detecting plant leaf diseases,” *Concurr. Comput. Pract. Exp.*, vol. 34, no. 28, p. e7357, Dec. 2022, doi: 10.1002/cpe.7357.

[265] T. Sultan *et al.*, “LeafDNet: Transforming Leaf Disease Diagnosis Through Deep Transfer Learning,” *Plant Direct*, vol. 9, no. 2, p. e70047, Feb. 2025, doi: 10.1002/pld3.70047.

[266] S. Ong, H. Ahmad, and A. H. A. Majid, “Development of a deep learning model from breeding substrate images: a novel method for estimating the abundance of house fly (*MUSCA DOMESTICA* L.) larvae,” *Pest Manag. Sci.*, vol. 77, no. 12, pp. 5347–5355, Dec. 2021, doi: 10.1002/ps.6573.

[267] C. Kaya, “Optimizing Crop Production With Plant Phenomics Through High-Throughput Phenotyping and AI in Controlled Environments,” *Food Energy Secur.*, vol. 14, no. 1, p. e70050, Jan. 2025, doi: 10.1002/fes3.70050.

[268] L. Xia, R. Zhang, L. Chen, L. Li, T. Yi, and M. Chen, “Monitoring the leaf damage by the rice leafroller with deep learning and ultra-light UAV,” *Pest Manag. Sci.*, vol. 80, no. 12, pp. 6620–6633, Dec. 2024, doi: 10.1002/ps.8401.

[269] J. You and J. Lee, “Offline mobile diagnosis system for citrus pests and diseases using deep compression neural network,” *IET Comput. Vis.*, vol. 14, no. 6, pp. 370–377, Sep. 2020, doi: 10.1049/iet-cvi.2018.5784.

[270] H. Li, Y. Jin, J. Zhong, and R. Zhao, “A Fruit Tree Disease Diagnosis Model Based on Stacking Ensemble Learning,” *Complexity*, vol. 2021, no. 1, p. 6868592, Jan. 2021, doi: 10.1155/2021/6868592.

[271] D. Sinanc Terzi, “Effect of different weight initialization strategies on transfer learning for plant disease detection,” *Plant Pathol.*, vol. 73, no. 9, pp. 2325–2343, Dec. 2024, doi: 10.1111/ppa.13997.

[272] L. Zhou, C. Zhang, F. Liu, Z. Qiu, and Y. He, “Application of Deep Learning in Food: A Review,” *Compr. Rev. Food Sci. Food Saf.*, vol. 18, no. 6, pp. 1793–1811, Nov. 2019, doi: 10.1111/1541-4337.12492.

[273] M. Grünig, E. Razavi, P. Calanca, D. Mazzi, J. D. Wegner, and L. Pellissier, “Applying deep neural networks to predict incidence and phenology of plant pests and diseases,” *Ecosphere*, vol. 12, no. 10, p. e03791, Oct. 2021, doi: 10.1002/ecs2.3791.

[274] M. Saberi Anari, “A Hybrid Model for Leaf Diseases Classification Based on the Modified Deep Transfer Learning and Ensemble Approach for Agricultural AIoT-Based Monitoring,” *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–15, Apr. 2022, doi: 10.1155/2022/6504616.

[275] S. Veluchamy, P. B. A, S. R. P. M, and S. D. S, “Paddy Crop Leaf Disease Classification Using Dense Kronecker Network,” *J. Phytopathol.*, vol. 173, no. 3, p. e70087, May 2025, doi: 10.1111/jph.70087.

[276] M. F. Danilevicz, S. R. Upadhyaya, J. Batley, M. Bennamoun, P. E. Bayer, and D. Edwards, “Understanding plant phenotypes in crop breeding through explainable AI,” *Plant Biotechnol. J.*, p. pbi.70208, Jun. 2025, doi: 10.1111/pbi.70208.

[277] A. Gutierrez, A. Ansuategi, L. Susperregi, C. Tubío, I. Rankić, and L. Lenža, “A Benchmarking of Learning Strategies for Pest Detection and Identification on Tomato Plants for Autonomous Scouting Robots Using Internal Databases,” *J. Sens.*, vol. 2019, pp. 1–15, May 2019, doi: 10.1155/2019/5219471.

[278] P. Bosilj, E. Aptoula, T. Duckett, and G. Cielniak, “Transfer learning between crop types for semantic segmentation of crops versus weeds in precision agriculture,” *J. Field Robot.*, vol. 37, no. 1, pp. 7–19, Jan. 2020, doi: 10.1002/rob.21869.

[279] V. Balaji *et al.*, “Deep Transfer Learning Technique for Multimodal Disease Classification in Plant Images,” *Contrast Media Mol. Imaging*, vol. 2023, pp. 1–8, May 2023, doi: 10.1155/2023/5644727.

[280] Z. Xu, H. Pan, W. Ye, Z. Xu, and H. Wang, “Detection method of wheat rust based on transfer learning and sharpness-aware minimization,” *Plant Pathol.*, vol. 72, no. 2, pp. 353–360, Feb. 2023, doi: 10.1111/ppa.13661.

[281] S. Aggarwal *et al.*, “Rice Disease Detection Using Artificial Intelligence and Machine Learning Techniques to Improvise Agro-Business,” *Sci. Program.*, vol. 2022, pp. 1–13, Jun. 2022, doi: 10.1155/2022/1757888.

[282] P. Preethi, R. Swathika, S. Kaliraj, R. Premkumar, and J. Yogapriya, “Deep Learning–Based Enhanced Optimization for Automated Rice Plant Disease Detection and Classification,” *Food Energy Secur.*, vol. 13, no. 5, p. e70001, Sep. 2024, doi: 10.1002/fes3.70001.

APPENDICES

Appendix 1: Function to Train and Test the Developed Model

```
n_epochs = 20

def train_model(model, criterion, optimizer, scheduler, num_epochs=n_epochs): # time
to train the model since = time.time()

# Initilize the weights of the best model

best_model_wts = copy.deepcopy(model.state_dict())

# initilize the best accuracy and epoch

best_acc = 0.0

best_epoch = 0

# storage of epochs, losses and accuracy

train_epochs, train_acc, train_loss = [], [], []

test_epochs, test_acc, test_loss = [], [], []

for epoch in range(num_epochs): # Iterate over the number of runs/epochs
```

```
print('Epoch {}/{}'.format(epoch, num_epochs - 1))
    print('-' * 10)
    # Each epoch has a training and validation phase
    for phase in ['train', 'test']:
        if phase == 'train':
            model.train()  # Set model to training mode
        else:
            model.eval()   # Set model to evaluate mode
        #initialize the loss.
        running_loss = 0.0
        #initilize correct predictions.
        running_corrects = 0
        # Iterate over data during each phase
```

```
for inputs, labels in dataloaders[phase]:
        inputs = inputs.to(device) #store in the GPU
        labels = labels.to(device) #store in the GPU
        # forward
        # track history if only in train
        with torch.set_grad_enabled(phase == 'train'):
            outputs = model(inputs) # predict outcome
            _, preds = torch.max(outputs, 1)
            loss = criterion(outputs, labels)
            # backward + optimize only if in training phase
            if phase == 'train':
                optimizer.zero_grad()
                loss.backward()
                optimizer.step()
```

```
    # statistics
    running_loss += loss.item() * inputs.size(0)
    running_corrects += torch.sum(preds == labels.data)
if phase == 'train':
    scheduler.step()
# find the loss and accuracy
epoch_loss = running_loss / dataset_sizes[phase]
epoch_acc = running_corrects.double() / dataset_sizes[phase]
# display the phase, loss and accuracy
print('{} Loss: {:.4f} Acc: {:.4f}'.format(
    phase, epoch_loss, epoch_acc))
# store the losses and accuracy depending on the phase
```

```
if phase == "train":
  train_loss.append(epoch_loss)
  train_acc.append(epoch_acc.item())
  train_epochs.append(epoch)
else:
  test_loss.append(epoch_loss)
  test_acc.append(epoch_acc.item())
  test_epochs.append(epoch)
# deep copy the model
if phase == 'test' and epoch_acc > best_acc:
    best_acc = epoch_acc
    best_epoch = epoch
    best_model_wts = copy.deepcopy(model.state_dict())
```

```
        print()
    #create a dictionary for metrics storage
    metrics_dict = {
        "losses" : {'train': train_loss, 'test': test_loss},
        "accuracy" : {'train': train_acc, 'test': test_acc},
        "epochs" : {'train': train_epochs, 'test': test_epochs},
    }
    # time the process
    time_elapsed = time.time() - since
    print('Training complete in {:.0f}m {:.0f}s'.format(
        time_elapsed // 60, time_elapsed % 60))
    print('Best test Acc: {:4f} @Epoch {:4f}'.format(best_acc, best_epoch))
    # load best model weights

model.load_state_dict(best_model_wts)
return model, metrics_dict
```

Appendix 2: Code for Generating the Confusion Matrix

```
names = ['common_rust', 'healthy', 'leaf_spot', 'nothern_leaf_blight']

def confusion_matrix(model, n_classes, names = names):

  # empty matrix of values

  matrix = np.zeros((n_classes, n_classes), dtype=int)

  # evaluate the model

  model.eval()

  # without gradient descent to through the test data to perform predictions.

  with torch.no_grad():

     for images, labels in dataloaders['test']:

       images = Variable(images)

       labels = Variable(labels)

       images, labels = images.to(device), labels.to(device)

       y = model.forward(images)

       pred = y.max(1, keepdim=True)[1]
```

```
for j in torch.cat((labels.view_as(pred), pred), dim=1).tolist():
        # actual and predicted
        matrix[j[0], j[1]] +=1
 correct = sum([matrix[i, i] for i in range(n_classes)])
 print("Correct predictions: ", correct, "of", dataset_sizes['test'])
 print(":----Confusion Matrix-----:")
 print(names)
 x = sns.heatmap(matrix, fmt='.0f', annot=True, xticklabels=names, yticklabels=names)
 x.set_title(f'Confusion matrix')
 return plt.show(x)
```

Appendix 3: Approval of Research Proposal and Supervisors

MURANG'A UNIVERSITY OF TECHNOLOGY

Office of the Director, Postgraduate Studies

E-mail: bps@mut.ac.ke
Website: www.mut.ac.ke

P.O. Box 75-10200
Murang'a, Kenya

Ref: MUT/ARP/PGS/19/2020 /VOL.1

Date: 6th June, 2024

Mr. Vincent Mbandu Ochago (SC500/5807/2022)
School of Computing and Information Technology,
Murang'a University of Technology,
P.O. Box 75-10,200
Murang'a.



Dear Mr. Ochago,

RE: APPROVAL OF RESEARCH PROPOSAL AND SUPERVISORS.

I am pleased to inform you that the Directorate of Postgraduate Studies on 15th May 2024 considered and approved your Doctoral research proposal entitled ***"Deep learning model for decision support of stakeholders in crop diseases and pest management"*** and appointed the following as supervisors:

1. Dr. Geoffrey Mariga -Murang'a University of Technology
2. Dr. Aaron Mogere Oirere - Murang'a University of Technology

You may now proceed with your data collection subject to obtaining research permit from NACOSTI, if required. You should also begin consulting your supervisors and submit through them quarterly progress reports to the Director Postgraduate Studies through your CoD and School Dean. Progress Reports forms can be accessed in the Student Portal.

It is the policy and regulations of the University that you observe deadlines. The guidelines on Postgraduate supervision can be accessed in the post graduate Handbook.

Your responsibilities as a student will include, among others;

i. Maintain regular consultation with your supervisor(s), at least once a month

ii. Submit quarterly reports on time, through your supervisors, CoD, Dean and to the Director of Postgraduate Studies;
iii. Ensure quality work all through;
iv. Present your research findings at 2-3 seminars/conferences prior to thesis examination.
v. Publish two articles from your research findings in a refereed journal prior to thesis examination.

For any further clarification please contact the undersigned.

Yours Sincerely,



Prof. Geoffrey Muchiri, PhD
Director, Postgraduate Studies
GM/PIM

Copy to:
1. Registrar – Academics and Students Affairs
2. Dean, School of Computing and Information Technology

Appendix 4: Approval Letter from NACOSTI

REPUBLIC OF KENYA

NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Ref No: **740683**

Date of Issue: **04/July/2024**

RESEARCH LICENSE



This is to Certify that Mr.. Vincent Mbandu Ochango of Murang'a University of Technology, has been licensed to conduct research as per the provision of the Science, Technology and Innovation Act, 2013 (Rev.2014) in Kiambu on the topic: A DEEP LEARNING MODEL FOR DECISION SUPPORT OF STAKEHOLDERS IN CROP DISEASES AND PEST MANAGEMENT for the period ending : 04/July/2025.

License No: **NACOSTI/P/24/37240**

740683

Applicant Identification Number

Director General
NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Verification QR Code

NOTE: This is a computer generated License. To verify the authenticity of this document, Scan the QR Code using QR scanner application.

See overleaf for conditions

Appendix 5: Publications

The following research papers were published from this thesis;

Journal Articles:

1. Vincent Mbandu Ochango, Geoffrey Mariga Wambugu, Aaron Mogeni Oirere, Deep Learning Model Factors Influencing Decision Support in Crop Diseases and Pest Management: A Systematic Literature Review, *International Journal of Information Technology and Computer Science(IJITCS)*, Vol.17, No.6, pp.52-69, 2025. DOI:10.5815/ijitcs.2025.06.03
2. Vincent Mbandu Ochango, Geoffrey Mariga Wambugu, & Aaron Mogeni Oirere. (2025). Comparative Analysis of Deep Learning Models for Crop Diseases and Pest Classification. *International Journal of Formal Sciences: Current and Future Research Trends*, *22*(1), 1–25.
3. V. M. Ochango, G. M. Wambugu, and A. M. Oirere. (2024). Deep Learning Model for Crop Diseases and Pest Classification. *International Journal of Computer and Information Technology*, (2279-0764), *13*(4), 138-151.
4. Vincent Mbandu Ochango, Geoffrey Mariga Wambugu, Aaron Mogeni Oirere, A CNN-SVM Hybrid Model with Radial Basis Function Kernel Optimization for Robust Crop Disease and Pest Classification. *International Journal of Engineering and Manufacturing* (Accepted for Publication)

Conference Papers:

1. Vincent Mbandu Ochango, Geoffrey Mariga Wambugu and Aaron Mogeni Oirere, Deep Learning Model Factors Influencing Decision Support in Crop Diseases and Pest Management: A Systematic Literature Review, in Gretsa University 7th International Multidisciplinary Research Conference, Monday 19th – Tuesday 20th August 2024.
2. Vincent Mbandu Ochango, Geoffrey Mariga Wambugu and Aaron Mogeni Oirere, A Hybrid Deep Learning Model for Crop Diseases and Pest Classification, in Krinyaga University 8th Annual International Conference, Wednesday 26th – Thursday 27th March 2025.