

AGRIGROW: A TECHNIQUE FOR CROP DISEASE DETECTION AND RECOMMEDATION

PRERIT KUMAR JAIN



Department of Computer Science and Engineering

Veer Surendra Sai University of Technology

Burla, Odisha-768018

2024-25

AGRIGROW: A TECHNIQUE FOR CROP DISEASE DETECTION AND RECOMMEDATION

**Project submitted in partial fulfillment of the requirement for the degree of
Bachelor of Technology in Computer Science and Engineering**

Submitted By:

PRERIT KUMAR JAIN

Regd. No. : 2102040011

Under the supervision of

Dr. Santi Kumari Behera



Department of Computer Science and Engineering

Veer Surendra Sai University of Technology

Burla, Odisha-768018

2024-25



Department of Computer Science and Engineering

Veer Surendra Sai University of Technology, Burla, Odisha

Date:

Certificate of Examination

Name: Prerit Kumar Jain

Registration Number: 2102040011

Title of the Project: **AGRIGROW: A TECHNIQUE FOR CROP DISEASE DETECTION AND RECOMMEDATION**

This is to certify that the project report entitled “**AGRIGROW: A TECHNIQUE FOR CROP DISEASE DETECTION AND RECOMMEDATION**” submitted by **Prerit Kumar Jain** Regd.no. **2102040011** of Computer Science and Engineering under the Department of Computer Science and Engineering, Veer Surendra Sai University of Technology, Odisha has been examined by us. We are satisfied with the volume, quality and correctness of the work.

External Examiners

HOD, Dept. Of CSE

VSSUT, BURLA



Department of Computer Science and Engineering
Veer Surendra Sai University of Technology, Burla, Odisha

Date:

Supervisor's Certificate

This is to certify that the Project work entitled “**AGRIGROW: A TECHNIQUE FOR CROP DISEASE DETECTION AND RECOMMEDATION**” is being submitted by **Prerit Kumar Jain** Regd.no. **2102040011**, to the Department of Computer Science and Engineering, Veer Surendra Sai University of Technology, Burla, in partial fulfilment of the requirement for the degree of Bachelor of Technology in Computer Science and Engineering during the academic year 2024-2025. It is an original report carried out by him under my supervision.

Dr. Santi Kumari Behera
(Supervisor)

Declaration

I hereby declare that the project titled “**AGRIGROW: A TECHNIQUE FOR CROP DISEASE DETECTION AND RECOMMEDATION**” submitted to Veer Surendra Sai University of Technology for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a result of original work carried out in this report. I also declare that the work has not been submitted, in whole or in part, to any other university as an exercise for a degree or any other qualification.

Date:
VSSUT, BURLA

Prerit Kumar Jain
Regd. No. 2102040011

Acknowledgement

I would like to thank my supervisor, **Dr. Santi Kumari Behera** for giving us the opportunity to undertake this project. I am extremely grateful for her insightful guidance, timely advice, continuous support, and motivation throughout my project work. I especially thank her for her kindness and excellent contributions. I would like to thank the HOD, Department of Computer Science and Engineering for his support to carry out my project work. I would like to thank all the faculty and staff members of the Department of Computer Science and Engineering at Veer Surendra Sai University of Technology, Burla for their help and support. I am deeply thankful to the Vice-Chancellor for his moral support and for providing us with all the necessary facilities during our candidature. I would like to thank our friends and family members for their unconditional love and support.

Name: Prerit Kumar Jain

Registration No.: 2102040011

Abstract

Global agriculture is confronted by the dual pressures of a burgeoning population demanding increased food production and the escalating impacts of climate change, which threaten crop yields and resource availability. To navigate these complexities, a shift towards precision agriculture, leveraging data-driven insights, is imperative. This major report details the design, development, and initial evaluation of an integrated digital decision support system aimed at empowering farmers and agronomists with actionable intelligence for enhanced crop management.

The first component is an **AI-powered crop disease detection module**. This module utilizes the capabilities of deep learning, specifically employing a pre-trained Convolutional Neural Network (CNN) based on the MobileNetV2 architecture. This lightweight yet powerful model is fine-tuned to analyse digital images of crop leaves and rapidly identify common diseases such as Cercospora Leaf Spot, Common Rust, and Northern Leaf Blight, or classify the plant as healthy. Early and accurate visual diagnosis through this automated system can significantly reduce reliance on manual inspection and facilitate timely intervention.

Secondly, the system incorporates an **intelligent crop recommendation engine**. This engine leverages machine learning models trained on datasets encompassing crucial soil parameters (Nitrogen, Phosphorus, Potassium levels, pH) and climatic factors (temperature, humidity, rainfall). By inputting site-specific data, users receive recommendations for the most suitable and potentially profitable crops, thereby optimizing land use and minimizing risks associated with planting ill-suited varieties.

The third component is a **tailored fertilizer recommendation system**. This system provides precise nutrient management advice by considering both the specific nutrient requirements of the selected or detected crop and the existing nutrient profile of the soil, as inputted by the user. The underlying machine learning model aims to suggest optimal quantities and types of fertilizers, promoting healthy plant growth while minimizing nutrient wastage, economic costs for the farmer, and potential environmental pollution from over-fertilization.

The entire system is implemented as an accessible web application. The backend services, including model inference and data processing logic, are developed using Python with the Flask microframework. The user interface, designed for intuitive interaction, is built using standard web technologies: HTML for structure, CSS for styling, and JavaScript for client-side dynamism. To ensure portability, ease of deployment, and environmental consistency, the application is containerized using Docker. Preliminary evaluations of each component indicate promising potential for enhancing agricultural decision-making, with the overarching goal of contributing towards increased crop yields, more efficient resource utilization, and the promotion of sustainable farming practices in a challenging global landscape.

Keywords: Crop Disease Detection, Crop Recommendation, Fertilizer Recommendation, Deep Learning, MobileNetV2

Contents

Certificate of Examination.....	I
Supervisor's Certificate.....	II
Declaration.....	III
Acknowledgement.....	IV
Abstract.....	V
List Of Figures.....	VII
List Of Tables.....	VIII
List Of Abbreviations.....	IX
1. Introduction.....	1
2. Literature Review.....	4
3. Methodology.....	9
3.1 About Dataset.....	9
3.2 About Machine Learning.....	10
3.3 About Deep Learning.....	12
4. Work Done.....	14
4.1 Data Loading and Preprocessing.....	14
4.2 Models Used.....	14
4.2.1 MobileNetV2.....	14
4.2.2 Random Forest.....	17
4.3 Bounding Box.....	19
4.4 User Interface and Frontend.....	20
4.4.1 Structure and Presentation.....	20
4.4.2 Interactivity and Client-Side Logic.....	21
4.5 Backend and Model Integration.....	22
4.5.1 Backend Integration.....	22
4.5.2 Deployment.....	23
5. Result and Discussion.....	24
5.1 Model Comparison with Variants.....	24
5.2 Model Comparison with Existing Models.....	25
5.3 Web Application Features.....	25
6. Conclusion and Future Scopes.....	29
6.1 Conclusion.....	29
6.2 Future Scopes.....	30
References.....	32

List of Figures	Page No.
Figure 1 Types of Crops (Tomato, Corn and Pepper)	10
Figure 2 Distribution of different labels in the train dataset	10
Figure 3 Architecture of MobileNetV2	17
Figure 4 Bounding Box Visualization for Pepper	20
Figure 5 Bounding Box Visualization for maize(corn)	20
Figure 6 Characteristic wise Bounding Box Height Histogram	21
Figure 7 Landing Page	23
Figure 8 Workflow Diagram	24
Figure 9 Services Offered	26
Figure 10 Crop Recommendation User Page	27
Figure 11 Crop Recommendation Result	27
Figure 12 Crop Disease Prediction User Page	28
Figure 13 Crop Disease Predicted Result	28
Figure 14 Fertilizer Recommendation User Page	28
Figure 15 Fertilizer Recommendation Result	29

List of Tables	Page No.
Table 1 All Previous Research Paper Works	7-8
Table 2 Model variant comparison	25
Table 3 Model comparison with existing models	26

List of Abbreviations

AI – Artificial Intelligence

ML – Machine Learning

DL – Deep Learning

CNN – Convolutional Neural Network

R-CNN – Region based Convolutional Neural Network

GDPR – General Data Protection Regulation

ANN – Artificial Neural Network

ReLU – Rectified Linear Unit

HTML – HyperText Markup Language

CSS – Cascading Style Sheet

JS – Java Script

JSON – JavaScript Object Notation

URL – Uniform Resource Locator

1. Introduction

Agriculture is the backbone of human life, sustaining, earning, and living for most of humankind. Yet the sector is also experiencing unprecedented challenges in the 21st century. An increasingly growing population of the world, projected to hit almost 10 billion by 2050, requires a mind-boggling increase in food production – an estimated 70% increase to the current level. To this need are being superimposed the rising imperatives of global warming, resulting in intractable weather aberrations, heightened pest and disease infestation, and loss of water and land resources. Conventional agriculture practices, which typically rely on intergenerational learning and human intuition, are unable to counter these amorphous challenges, thereby resulting in low production, loss of resources, and environmental stress. These are multi-faceted problems demanding a paradigm change to more accurate, data-driven, and sustainable agriculture.

The advent of new technologies, particularly artificial intelligence (AI), Machine Learning (ML), and Data Science, can revolutionize agriculture. Precision agriculture, backed by these technologies, endeavours to optimize farming practice and input at a minute level, generating maximum output with minimum waste and environmental degradation. On the back of the potential of data analytics and intelligent algorithms, I can move beyond personalized recommendations to achieve site-specific, condition-specific management practices to allow farmers to make more informed decisions.

This project fits within the general mandate of the Responsible Artificial Intelligence (AI) Lab (RAIL) of the Kwame Nkrumah University of Science and Technology in Ghana. RAIL's purpose is to develop responsible AI research and innovation to meet Africa's development demands. With a focus on multidisciplinary collaboration and women's involvement in AI, RAIL operates with a vision of establishing strong local capacities and making the ethical use of AI available to agriculture, biomedicine, and environmental science. With ethics-centered capacity development in AI, the laboratory is endeavouring to provide a positive impact to marginalized societies in Ghana and Senegal and beyond the sub-region. Other than that, it also intends to put research in AI at the global forefront.

RAIL is part of the broader AI4D Africa, which is a 10-year collaboration between the Foreign Commonwealth Development Office (FCDO) and the International Development Research Centre (IDRC). RAIL complements policy, innovation, and leadership engagements with the ultimate aim of instilling improvement in the quality of life of the continent.

Project goal directs design, development, and testing of an integrated system to address the key decision points of the life cycle of agriculture. Adjusting to the dependent nature of the crop's diet, fit, and well-being, three significant components are integrated into the project: (1) Crop Disease Detection system, (2) Crop Recommendation engine, and (3) Fertilizer Recommendation system. The common objective is to offer farmers a complete digital platform for backing up potential crop health threats, choosing the most suitable crops to their quite distinct environmental conditions, and optimizing nutrient care for optimal growth and yield.

The **Crop Disease Detection module** takes advantage of the power of deep learning, in this case Convolutional Neural Networks (CNNs), to analyse digital images of crops (first pepper and tomato crops, later corn) and automatically detect whether and what type of epidemic disease is present. Early detection and precision are essential in order to offer early intervention in order to avoid mass outbreaks and reduce possibly catastrophic yield loss. This approach aims to decompose problems of human examination, which tends to be slow, subjective, and based on a body of knowledge that is not available.

The **Crop Recommendation system** solves the most critical issue of selecting crops. The crops which are not acceptable to the existing soil, climatic regimes (rainfall and temperature), and other environmental factors are the major source of inefficient productivity and wastage of resources. The system is created based on machine learning software that learns from vast databases of soil factors (pH, nutrient content, texture), climatic properties (mean temperature, rainfall), and crop historical yields. After the inputs are given by region, the system suggests the most suitable and likely to be profitable crops, scheduling farm activities according to ecological appropriateness.

Along with the first two considerations arises the **Fertilizer Recommendation system**. Ideal growth of crops arises due to ideal availability of macronutrients. Under-fertilization (leading to poor growth and decreased yield) and over-fertilization (leading to environmental pollution in the form of runoff, soil pollution, and undue farmer expenditure) are not optimal. The system calculates soil nutrition factor critical values (such as Nitrogen, Phosphorus, Potassium content) and the nutrition requirements of the chosen or accepted crop with a view to offering the correct fertilizer suggestions. This assists in supplying crops with the correct nutrients at the correct amount for sustainable growth, achieving maximum production capacity, and encouraging healthy practice of soil maintenance.

The detailed objectives for each crop are as follows:

1. Tomato

Tomatoes are widespread vegetables throughout the world and are susceptible to some diseases like tomato leaf curl virus, early blight, and late blight. These three diseases spread through colour change in leaves, curling, and leaf spotting, resulting in huge loss of yield as well as economic loss. The initiative is focused on creating machine learning models to analyse images of tomato leaves in an effort to identify such diseases with high accuracy. The models also need to be able to tell the difference between healthy and diseased leaves, a factor that would allow farmers to act in whatever way possible in an effort to salvage their crops. Second, the answer has to be generalizable sufficiently to be able to generalize to the majority of kinds of tomatoes and conditions in an effort to make it applicable under a variety of conditions for agriculture.

2. Corn

Maize is among the safest income-generating livelihood-supported food staple crops sustaining livelihoods for millions of farmers around the world. It is also very susceptible to diseases like northern leaf blight, grey leaf spot, and common rust, which most of the time cause a drastic decline in yields. Diseases manifest themselves in lesions, discoloration, and necrotic spots on the leaves. The task for corn would be to develop machine learning models that can identify the diseases properly from the pictures of corn leaves and differentiate between more than one disease per leaf, if the situation would be viable. The models must also be able to handle environmental variance, like light and background change, to provide accuracy in actual situations.

3. Pepper

Pepper is a very lucrative crop for food and financial returns. Pepper is mostly susceptible to diseases such as pepper blight, powdery mildew, and anthracnose that result in fruit rotting, leaf spots, and plant shortening. In this case, the aim is to develop machine learning algorithms that can identify these diseases at their initial stages and diagnose the health status of pepper crops accurately. Since peppers are economically important to smallholder farmers, the models must be light and mountable on hand-held equipment such that they become feasible for real-time monitoring in the field.

2. Literature Review

Chakraborty and Newton et al. (2011) have studied the complex relationship between climatic change and crop disease occurrence. They noted how altered climatic patterns, rising temperatures, and elevated humidity levels provide the ideal breeding ground for disease propagation. This study grasped the dynamic nature of crop diseases and the need for machine learning models that would be able to learn and adapt to such evolving environmental patterns. With climate-resilient models in their arsenal, they crossed the barrier of using environmental inputs to identify crop disease, removing uncertainty regarding true conditions[1].

Deep convolutional neural networks (CNNs) potential to distinguish agrarian diseases was brought forth by **Mohanty et al.** (2016). Their model, when they had trained it on a set of 26 diseases on 14 crops, had an impeccable accuracy of over 99% on controlled sets. This was initial evidence in demonstrating that CNNs would have the capacity to diagnose disease with truly unmatched levels of precision. But this crippling limitation held when the model was subjected to real-world field imagery, and accuracy disintegrated. That discrepancy pointed out focus on models needing to generalize over to novel and new situations. Mohanty et al.'s research is still the benchmark in the literature, proof of the necessity to close the gap between laboratory findings and field application[2].

Sladojevic et al. (2016) wrote generalization of disease detection models through transfer learning. Applying pre-trained CNNs and adapting them for a specific crop disease, they improved the capability of their models to generalize to novel disease. This approach, apart from reducing dependence on massive sets, also rationalized training processes. Their study reinforced the argument that transfer learning has the potential to provide scalable solutions to the long-range challenges dominating the detection of crop diseases and that it is a break through towards accommodating machine learning systems[3].

Amara et al. (2017) used CNNs in labelling banana leaf disease with a very high accuracy rate in laboratory conditions. Even though they performed optimally in laboratory conditions, models failed to function their role in sunlight variations, perspective, and environment in actual conditions. Models' susceptibility in actual conditions exhibited how it had to be realized to obtain substitute and representative datasets to make models robust. Amara et al.'s work depicted overcoming data sparsity and environmental diversity in Agri-machine learning studies[4].

Fuentes et al. (2018) opened the door to symptomatic disease-detecting crop disease detection object detection models like Faster R-CNN. Compared to conventional classification models, their work localized and annotated symptomatic disease images, which would enable farmers to take optimal decisions. Their research established the contribution of combining detection and classification in precision farming such that appropriate interventions can be implemented. This research was a breakthrough in increasing application in real-time since it not only facilitated disease detection for farmers but also yielded the ability to determine where source points of infection are, enhancing the utilization of resources and disease control[5].

Ramcharan et al. (2019) emphasized a mobile-phone-based system for the diagnosis of disease that was uniquely designed for low-input farmers. With the use of a deep learning model being included in the smartphone application, they were able to provide a level of accuracy of 80% under true field conditions. The convenience and usability aspect of its use, coupled with a simple-to-use interface, were the particular areas of application of this paper to the smallholder farmers who possessed very little technical knowledge. Their focus was on the potential of mobile technology to put sophisticated diagnostic capability in the hands of the masses, where plant disease diagnosis and management are enhanced for farmers more[6].

Too et al. (2019) countered the absence of small and thin datasets with rotation, zooming, and flipping data augmentation techniques. They artificially augmented their training data and thus the capability of their CNN model to identify disease in tomatoes, as well as other vegetables and fruits. The focus was on the importance of data contribution by augmentation towards reducing the lack of data limitation and restriction, and further improving model quality and robust detection in its practical application to agriculture[7].

Zhang et al. (2020) evaluated the multi-task learning model to address the problem of concurrent crop diseases. In their experiment, they could identify several diseases from a single image, i.e., pepper and tomato plants, using feature transfer across tasks. Their multi-task model improved classification and diagnosis performance, far more akin to actual farm environments where crops may be infected with multiple diseases simultaneously. Zhang et al.'s study showed that it was possible and worth doing to develop holistic models for complex agricultural issues[8].

David et al. (2021) described the real-world consequences of running machine learning models on edge devices like smartphones. With TensorFlow Lite, they simplified CNN architectures to be easily compressed into both computational as well as memory requirements at the expense of no loss in accuracy. The research recognized farmers' misconception in creating environments where advanced disease detection technology would be able to continue running

and deploy on low-end devices. David et al. research associated high-tech and low-tech to allow smallholder farmers to gain capability for real-time diagnostics[9].

Ma et al. (2021) attempted to improve CNN models with pruning techniques for the construction of light-weight models deployable on low-power devices. The method achieved a balance between model performance and computation cost for real-life deployment by smallholder farmers. The study provided significant contributions towards the construction of efficient machine learning models for limited environments, making agricultural technology more accessible[10].

Table 1: All Previous Research Paper Works

Sl. No.	Author	Year	Outcome
1	Chakraborty and Newton et al.	2011	Recognized that altered climatic trends and rising temperatures enable the spread of diseases, where machine learning models which can be modulated based on environmental fluctuations are required. Their research laid the foundation for the incorporation of climate information into crop disease monitoring systems.
2	Mohanty et al.	2016	Reached over 99% accuracy with the assistance of constrained test data and the ability to use CNNs for application in disease diagnosis. But efficiency fell significantly on real images, mirroring the demands of generalization models to other field conditions.
3	Sladojevic et al.	2016	Increased ability to learn from new diseases and reduced reliance on the presence of large datasets through the use of fine-tuned pre-trained CNNs. The model showcased the scaling of machine models to other agronomy applications.
4	Amara et al.	2017	Exhibited high accuracy in laboratory settings but did not generalize to field settings. The research reaffirmed that good models require big, heterogeneous datasets.
5	Fuentes et al.	2018	Localized and distinguished disease symptoms from pictures with actionable data for specific intervention. It was demonstrated to work well effectively enough for precision use in agriculture.

Sl. No.	Author	Year	Outcome
6	Ramcharan et al.	2019	Worked at a performance of approximately 80% correctness under real-world conditions. The project focused on enabling the usage of the system as well as the availability thereof, and that capability of empowering the farmer to augment diagnostic ability via the mobile app.
7	Too et al.	2019	Improved the robustness and accuracy of CNN models for tomato disease detection. This work highlighted the critical role of data augmentation in enhancing model performance.
8	Zhang et al.	2020	Improved performance on diagnosis and classification efficacy through sharing of features across tasks. The research proved that multiple diseases can indeed be diagnosed contemporaneously in advanced agriculture settings.
9	David et al.	2021	Developed thin models that were highly accurate and required less computation and memory. All this made the process easier for farmers on low-end phones in developing economies.
10	Ma et al.	2021	Achieved very high accuracy of classification with sudden drops in model size and complexity. The contribution of the research was pragmatic to the use of machine learning as a solution to smallholder farming.

3. Methodology

3.1 About Dataset

These facts were gathered from farms and forests in Ghana and this was all centered on three simple crops, i.e., pepper, tomato, and corn. These are big crops to the economy and food security of this country but because they are prone to some diseases, their productions are of smaller scale. The dataset was built in order to train machine learning algorithms that are able to detect and classify diseases in these crops such that there exists a foundation for scalable, real-time diagnostic platforms for empowering farmers to better confront crop health challenges as seen in figure 1.

The data set is comprised of thousands of high-resolution crop leaf photographs captured under varying natural conditions to strengthen the machine learning models. Each photograph has been manually annotated by agriculture experts, labelling if the leaf is healthy or infected with a specific disease. Proper annotation makes the data set reliable and suitable for developing highly precise disease detection models. Key diseases in the dataset are tomato leaf curl virus, pepper blight, early blight, late blight, and maize leaf spot. Discoloration, necrotic spots, curl, and lesions are symptoms well illustrated here by which machine learning algorithms are trained to detect infected plant samples and normal samples as illustrated in figure 2.

One of the most significant advantages of this dataset is that it is heterogeneous. It contains images with different lighting, angle, and background, mimicking real-world difficulties faced by farmers. Accounting for these environmental differences, the dataset guarantees resulting machine learning algorithms to be robust and capable of generalizing well across contexts. This is especially important in countries such as Sub-Saharan Africa, where the farm labourer's work under conditions of unbalanced light and capricious weather conditions.

The other critical feature of the dataset is also the fact that it aligns with the vision of running machine learning models on the likes of low-end phones. In the sense that Sub-Saharan farmers widely use low-end phones for communication as well as for work, the dataset has been designed to facilitate support for offering support for the generation of light-weight models with lower power usage. Collection of datasets was a daunting task with farm farmers and scientists. Forestland and plantation farm images were captured in Ghana to obtain domesticated plants as well as wild populations that could have early indications of disease. Aside from this, it not only gives a glimpse of the occurrence of disease but also allows early

detection, which can be beneficial in the prevention of outbreaks prior to transmission being feasible.

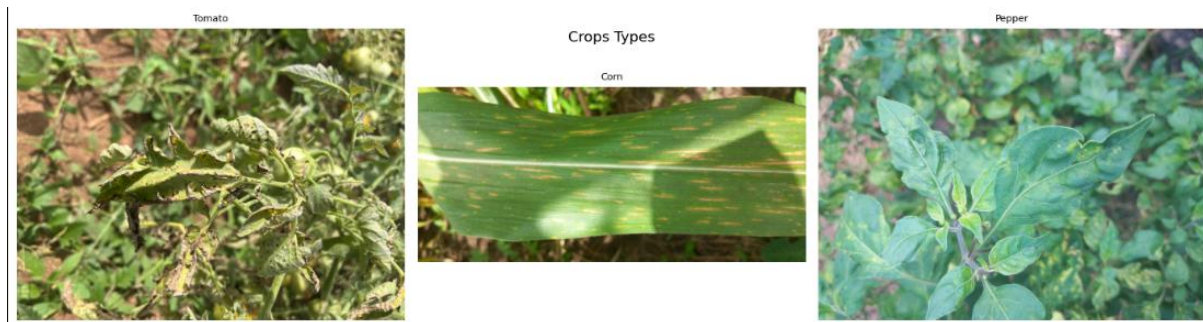


Fig. 1 Types of Crops (Tomato, Corn and Pepper)

Distribution of different labels in the train dataset

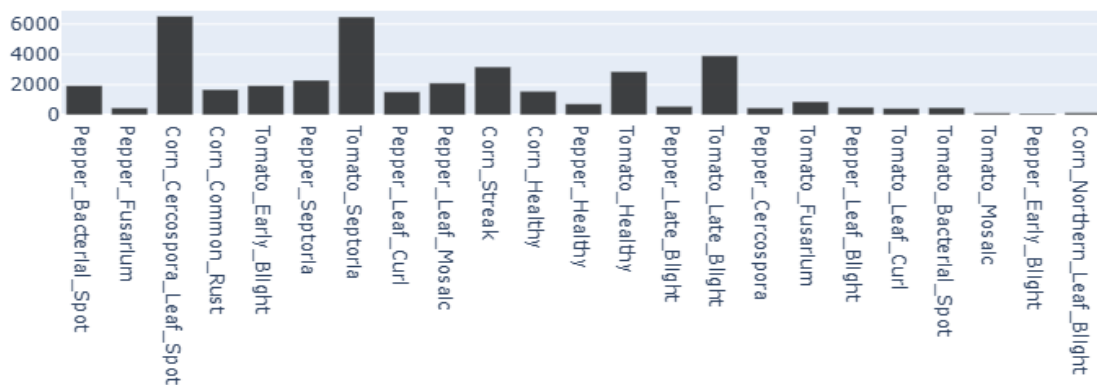


Fig. 2 Distribution of different labels in the train dataset

3.2 About Machine Learning

Machine learning (ML) is an AI subset which enables machines to learn and improve by themselves from their own experience without hand-coded programming. The machine learning algorithm can learn things and learn patterns and expertise and process things at an automatic level by monitoring relation and pattern between information. With the ability of continuous revision and modifications, its contribution to superior decision-making rendered machine learning a valuable technology across most sectors such as medicine, moneymaking, transports, and cultivation.

Essentially, machine learning is based on three general paradigms of learning: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning learns a model from labelled data whose output is specified. Spam or not spam email categorization, home price estimation, or object detection from pictures are just some illustrations.

Unsupervised learning works on unlabelled data. Algorithms in this classification, such as clustering and dimensionality reduction, uncover implicit patterns and structure of the data. Lastly, reinforcement learning learns to behave in an environment to optimize cumulative reward, used in robotics and playing games AI like AlphaGo.

The most prominent benefit of machine learning is that it can be implemented across different applications. In the field of medicine, for instance, ML programs provide disease diagnosis, patient prognosis, as well as personalized treatment plans to the individual. Machine learning algorithms that are trained on medical images can recognize abnormalities such as tumours much more accurately than the radiologists, freeing them of a colossal workload. Similarly, in the field of finance too, ML is utilized for detecting dubious transactions, assessing credit risk, and implementing automated trading rules so as to ensure optimum efficiency and not face the possibility of human error.

The transport sector also has been revolutionized by machine learning. Autonomous cars utilize deep learning, which is a form of ML, to analyse sensor data and make real-time decisions for highway driving and safety. Ride-sharing companies like Uber and Lyft use ML code to determine best routes, predict demand, and dynamically price rides. In logistics, ML improves supply chain management by forecasting demand, monitoring inventory levels, and estimating route delivery in order to reduce costs and increase customer satisfaction.

Machine learning agriculture eliminates grand obstacles such as food security, utilization of resources, and resistance to weather. Precision farming employs ML to observe the condition of the ground, forecast yields, and analyse disease or insect infestation in real time. For example, image recognition software can track satellite or aerial images to determine areas of land destruction in fields and enable farmers to take action at a point. Such intervention based on data achieves maximum productivity while reducing the ecological impact of agriculture.

Though being extremely successful, machine learning has some issues that need to be resolved by researchers and practitioners. Amongst these, perhaps the biggest are the quality and quantity of data. Good-performing ML models require enormous sets of unique, high-quality data to train on. In most instances, however, data will instead be sparse, missing, or biased, and thus result in spurious or incorrect predictions. The second amongst these is data privacy, and that is primarily where the data is sensitive, i.e., health, finance. It is hard to balance compliance with regulation like the General Data Protection Regulation (GDPR) without compromising model performance.

The third one is explaining and interpreting machine learning models. While smaller models, such as linear regression, provide interpretable decisions, more sophisticated models, such as deep neural networks, are often referred to as "black boxes."

3.3 About Deep Learning

Deep Learning (DL) is a powerful and emerging branch of Machine Learning (ML), which is itself a branch of Artificial Intelligence (AI). It's an innovative idea that significantly differs from traditional approaches to ML by using artificial multi-layered neural networks (thus "deep") to learn sophisticated patterns and hierarchical representations from raw data. Loosely motivated by the structure of interconnected neurons in the human brain and the structure and functioning of the human brain, DL models have been wonderfully successful across an amazingly wide variety of tasks, revolutionizing areas such as computer vision, natural language processing, speech recognition, and recommendation systems.

Deep Learning is based on Artificial Neural Networks (ANNs). An ANN consists of processing units that are connected to each other and called artificial neurons or nodes, arranged in a variety of layers usually: an input layer, some hidden layers, and an output layer. Each connection between the neurons has a weight, which determines the signal passing through it. Every neuron receives multiple inputs from different connections, calculates the weighted sum of the inputs, adds a bias, and feeds the output to a non-linear activation function (e.g., Sigmoid, Tanh, ReLU - Rectified Linear Unit) to produce an output. The network needs non-linearity so that it can represent complex, non-linear relationships in the data.

The "deep" characteristic is the presence of many hidden layers placed between the output and input layers. Standard neural networks would consist of one or possibly two hidden layers, but deep architecture may have thousands, hundreds, or tens of layers. Depth allows the model to learn a hierarchy of features or representations. Early layers can be trained to detect low-level features like edges, corners, or textures from the raw input data itself (e.g., pixel intensity values in an image). Higher levels utilize these low-level features and combine them to learn more and more complex patterns (e.g., shapes, components of an object), and even lower levels combine these patterns and add them to recognize abstract objects or whole objects (e.g., a particular type of leaf, a particular type of disease lesion).

DL training is typically acquired through training in DL with a process called the backpropagation algorithm with a process called optimization. For each input in a labelled set employed during training, the input is passed forward through the network to produce an output

estimate. This prediction is then compared to the actual ground truth label using a loss function (cross-entropy in classification, mean squared error in regression) that calculates the difference or error. Then backpropagation employs an algorithm to compute the gradient of that loss function with respect to all the weights and biases in the network, propagating backwards from output layer to input layer.

These gradients specify by how much each parameter needs to be adjusted so as to decrease the error. The optimizer uses these gradients to modify the weights and biases of the network in infinitesimally small amounts by repeating many iterations many times on a large training set (epochs) with each iteration repeatedly making the model better and better at predictions.

Deep Learning, and the use of CNNs in particular, provides the most appropriate platform for the Crop Disease Detection feature for the project's requirements. Identification of plant leaf images for indications of hidden visual signals of diseases – i.e., colour variation, texture variation, spot morphology, and lesion patterns – is a problem best addressed by CNNs' hierarchical feature learning ability. By learning on a large database of labelled images of corn leaves from a healthy crop and any severity of disease, a deep CNN model has the capability to learn automatically to detect each condition's subtle visual features for making rapid and even more precise than with the human eye diagnoses.

4. Work done

4.1 Data Loading and Preprocessing

Data loading and initial visualization are critical early phases in machine learning, where raw data are converted into informative information. The phases determine the features of the data, identify outliers, and develop an intuitive understanding of underlying information prior to advanced analytical processes.

Preprocessing data is an important preparation step in machine learning that transforms raw data to clean, consistent, and model-formatted data. The most important step is missing value treatment, duplicates removal, feature normalization or scaling, and encoding category variables to prepare the data to be improved and enhanced in model quality. Through overcoming flaws in data and input feature normalization, data preprocessing aids machine learning algorithms to understand data more effectively, reduces bias, and significantly enhances the power to make useful and reliable inferences from intricate data.

4.2 Models Used

4.2.1 Deep Learning Model for Crop Disease Detection: MobileNetV2

To enable independent disease detection of plants from leaf images, the present work utilized a deep learning approach in the form of a Convolutional Neural Network (CNN) solution. A pre-trained **MobileNetV2** architecture was selectively utilized and heavily fine-tuned for task. MobileNetV2 was selected especially because of the embedding of its natively structural strengths, its proven image classification capability, and its applicability to active environments refer the architecture shown in figure 3. The previous mentioned upgraded model, having undergone rigorous trials over a specific test set of multiple hundred images relating to various classes of diseases in major crops like corn, tomato, and pepper, was found to be **91%** correct on average to distinguish the diseases from one another and from the samples of healthy plants.

Rationale for Selecting MobileNetV2:

Some of the reasons for selecting MobileNetV2 over other feasible CNN architectures :

Computational Efficiency and Mobile/Edge Deployment Friendliness: MobileNetV2 is tailored towards low-resource environments. Its architecture incorporates advancements in the form of depth wise separable convolutions and inverted residual blocks with linear bottlenecks. These traits significantly reduce the number of parameters and computations compared to larger, more traditional models without a corresponding trade-off in accuracy. Such efficiency is necessary for deployment in the future on mobile phones for in-field use by farmers or on

low-power edge computing hardware, where computational power and battery life are constrained. Even in this current web-deployment, the efficiency translates into less waiting time for users to carry out inference and reduced server run costs.

Strong Performance on Large-Scale Image Classification Benchmarks: Despite being small in size, MobileNetV2 has attained competitive accuracy on large-scale image classification benchmarks like ImageNet. Such proven performance demonstrates its strong capacity to learn strong and discriminative visual features, which is essential to differentiate the typically subtle visual cues of various plant diseases – e.g., textural changes, discoloration, or leaf spots.

Capitalizing on Transfer Learning to Achieve Enhanced Performance and Increased Efficiency: This project utilized pre-trained MobileNetV2 that was trained using the enormous ImageNet dataset with over a million diverse images of a thousand common object classes. Transfer learning helps system tap into the high-density, hierarchical visual features (e.g., edges, texture, shapes, simple object pieces) learned by the model through the large database. By fine-tuning this pre-trained model on my own specific crop disease data, I achieved the reported over 90% accuracy using a relatively smaller domain-specific dataset and with much less training time than would be required to train an equally deep CNN from scratch. These pre-learned general features are highly transferable and provide an excellent starting point for the task of identifying disease symptoms on plant leaves.

Ideal Balance Between Model Size and Predictive Accuracy: While some extremely large models might eke out slightly higher accuracy scores on some tasks, they always do so at the expense of a much larger model footprint and increased computational demand. MobileNetV2 achieves an excellent and practical compromise by providing stable predictive accuracy (as evidenced over 90% in this study) with a reasonable model size suitable for real-time use where deployment overheads and processing speed are essential considerations.

Ease of Implementation within Standard Frameworks and Availability of Pre-trained Weights: Good pre-trained MobileNetV2 models are readily available in popular deep learning frameworks like TensorFlow/Keras, used in this project. This ease of access facilitated the process of implementation, allowing the project to focus on key areas such as dataset collection and enrichment, designing an effective fine-tuning strategy, and integrating the model with the overall application, rather than wasting resources on designing and extensively training an advanced CNN architecture from scratch.

Fine-tuning Process: The fine-tuning of the pre-trained MobileNetV2 for specific crop disease classification task involved a series of significant steps. To start, the first top classification layer of the ImageNet-trained MobileNetV2 (which is trained for 1000 classes) was removed. It was replaced with a new chain of layers, often including a global average pooling layer and a chain or multiple of dense (fully connected) layers where the final dense layer utilizes a SoftMax activation function specialized for the number of the classes of diseases in each of the crops contained within the database (for example, 4 classes in the case of corn: Gray Leaf Spot, Common Rust, Northern Leaf Blight, and Healthy). During the initial phase of fine-tuning, the MobileNetV2's base convolutional weights were frozen and the extra classifier layers were trained only. This allowed the new layers to pick up from the features learned by the pre-trained base. Afterwards, a percentage of the remaining subsequent convolutional layers in the MobileNetV2 base were "unfrozen," and the entire model was trained end-to-end at an extremely low learning rate. This method allows the pre-trained features to be sub-tirely adapted to my own specifics of specific crop disease images. The standard data augmentation techniques of random rotation, horizontal flip, and small brightness adjustment were also applied to the training images to increase dataset diversity and generalize the model's ability to perform well on new, unseen images.

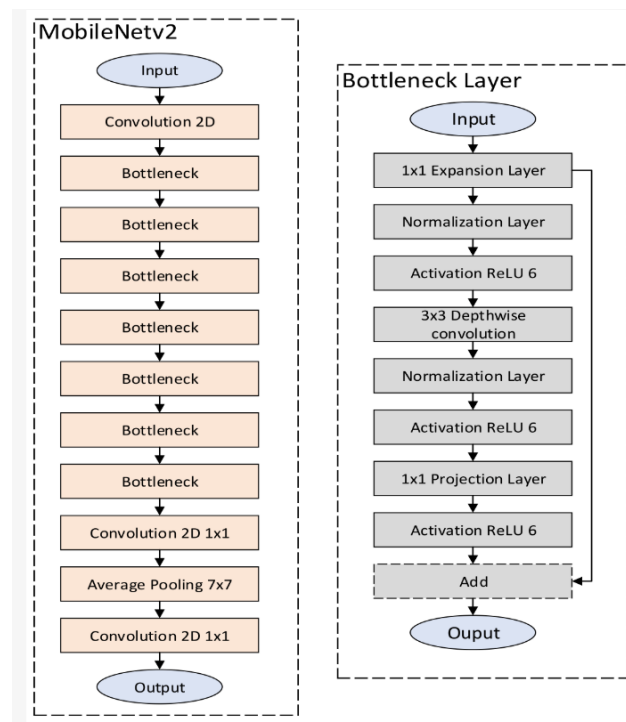


Fig 3. Architecture of MobileNetV2

4.2.2 Random Forest Model for Crop and Fertilizer Recommendation

For the intelligent crop recommendation module of this agricultural decision support system, a **Random Forest algorithm** was applied. Random Forest is a powerful and general supervised learning algorithm of the ensemble family. Random Forest works by creating an enormously large number of decision trees while training and predicting the class that is the mode of the classes (classification) or average prediction (regression) of the trees. In our scenario, it is applied to classification – to anticipate the best crop from a list given.

Random Forest Working Process:

Bootstrap Sampling (Bagging): Random Forest constructs many decision trees. Each tree, it takes a random sample of training data with replacement (bootstrapping). Here, some of the data points are used many times in a single tree training set, while others are not used at all.

Random Feature Selection: In every node of a decision tree, while selecting the optimal split, Random Forest never looks at all the features available. Instead, it selects a random subset of features and selects the best split from that subset alone. The number of features to look at per split is a hyperparameter.

Growing Multiple Decision Trees: Steps 1 and 2 are iterated in order to grow a large number of different decision trees (the "forest"). Each tree is grown to full depth without pruning (although sometimes pruning can be used).

Aggregation for Prediction: When a new data point (e.g., soil N, P, K, temperature, rainfall values) must be classified, it is propagated down each tree in the forest. Each tree "votes" for a class (a particular crop). The Random Forest algorithm then chooses the most voted class as the final prediction.

Reasons for Using Random Forest for Crop Recommendation:

The selection of Random Forest for the task of crop recommendation was made due to various favourable properties of the algorithm:

High Accuracy and Robustness: Random Forests are less likely to overfit and are highly accurate in their predictions. They perform very well on most types of datasets by averaging the predictions of numerous individual trees (reduction of variance) and are less likely to overfit than a single decision tree, particularly when the number of trees is sufficiently high. This is important in order to give reliable crop suggestions.

Handles Non-Linear Relationships Well: The interaction among soil parameters, climatic factors, and ideal crop selection is usually non-linear. Decision trees, and hence Random

Forests, are well-suited to capture the non-linear relationship without the need for special feature engineering (e.g., polynomial features).

Handling High-Dimensional Data: Random Forests can handle data with many input features (although your current feature set for crop suggestions may be of moderate size). Random selection of a feature at every split makes it easy to handle high dimensionality.

Good Performance with Default Hyperparameters: Random Forests can produce good performance using default hyperparameters and minimal or no hyperparameter tuning and are therefore easy to implement and use. Hyperparameter tuning will certainly improve performance, but they are less sensitive to hyperparameter choice than certain more complex models.

Implicit Feature Importance Ranking: Random Forest has a built-in method of determining the importance of every input feature to prediction (e.g., Gini importance or permutation importance). This can be extremely informative towards determining what variables (e.g., rainfall, soil N, temperature) have most impact in deciding on crop suitability, which is worthwhile information in itself irrespective of the recommendation.

Handles Missing Data and Outliers Reasonably Well: Not immune, but the ensemble character can make Random Forests relatively more robust against missing values (if properly dealt with during tree construction or imputation) and outliers than some other algorithms.

Effective for Training and Prediction (for comparatively smaller-sized datasets): Although training masses of trees is computationally demanding in highly large datasets, for usual dataset sizes encountered by such recommendation problems, Random Forests are effective. Even prediction is quick because it entails traversing pre-trained trees for input.

Good for Categorical and Numerical Data: Random Forests are particularly well-suited to deal with a combination of numerical features (e.g., NPK values, temperature) and categorical features.

4.3 Bounding Box Visualization

Bounding box visualization is a simple technique in computer vision and object detection that graphically depicts where and how big the objects detected are in an image. Through the visual description of rectangles enclosing the detected objects, the method provides a simple, easily interpretable way of displaying machine learning model predictions so that researchers and developers can simply verify the correctness of object detection algorithms as shown in figures 4 and 5, height histogram for characteristic of crops can be plotted as shown in figure 6.

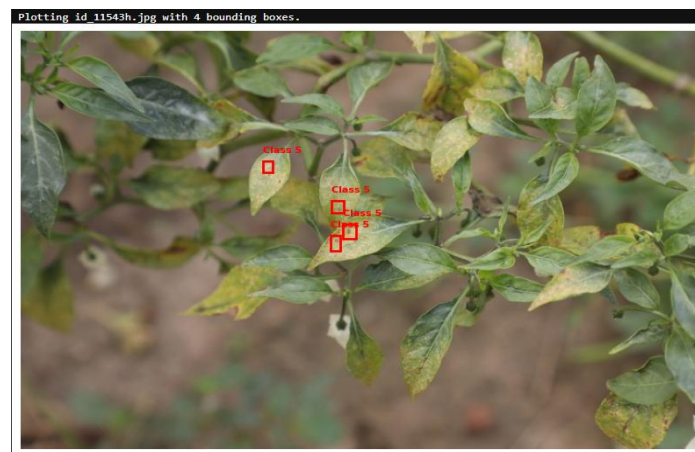


Fig. 4 Bounding Box Visualization for Pepper

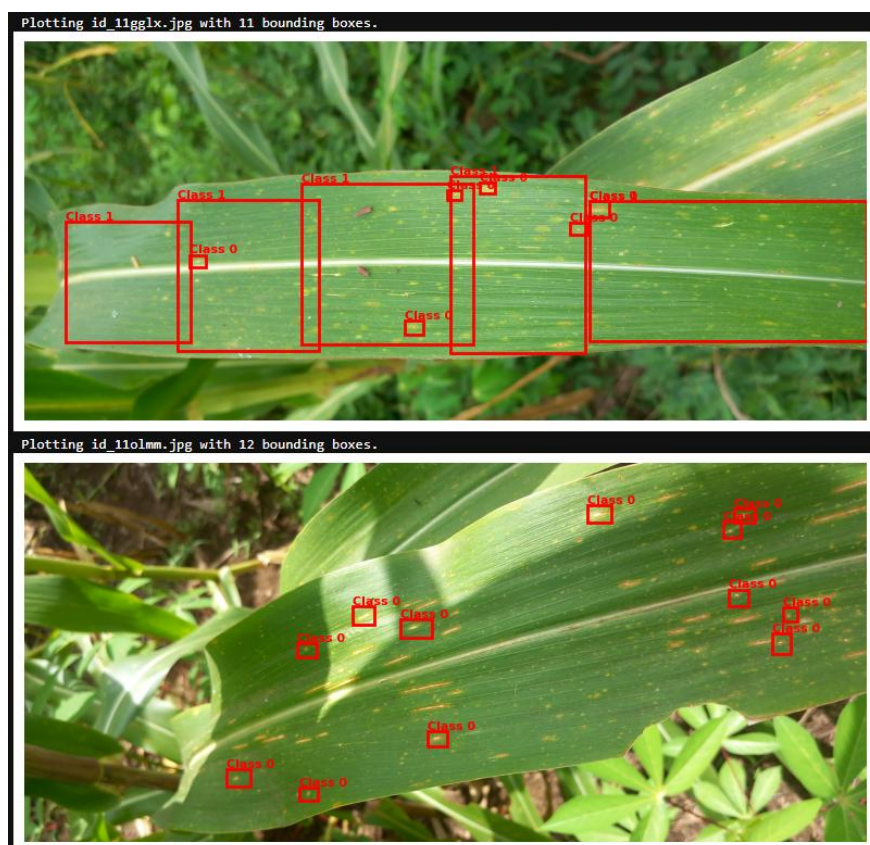


Fig. 5 Bounding Box Visualization for maize(corn)

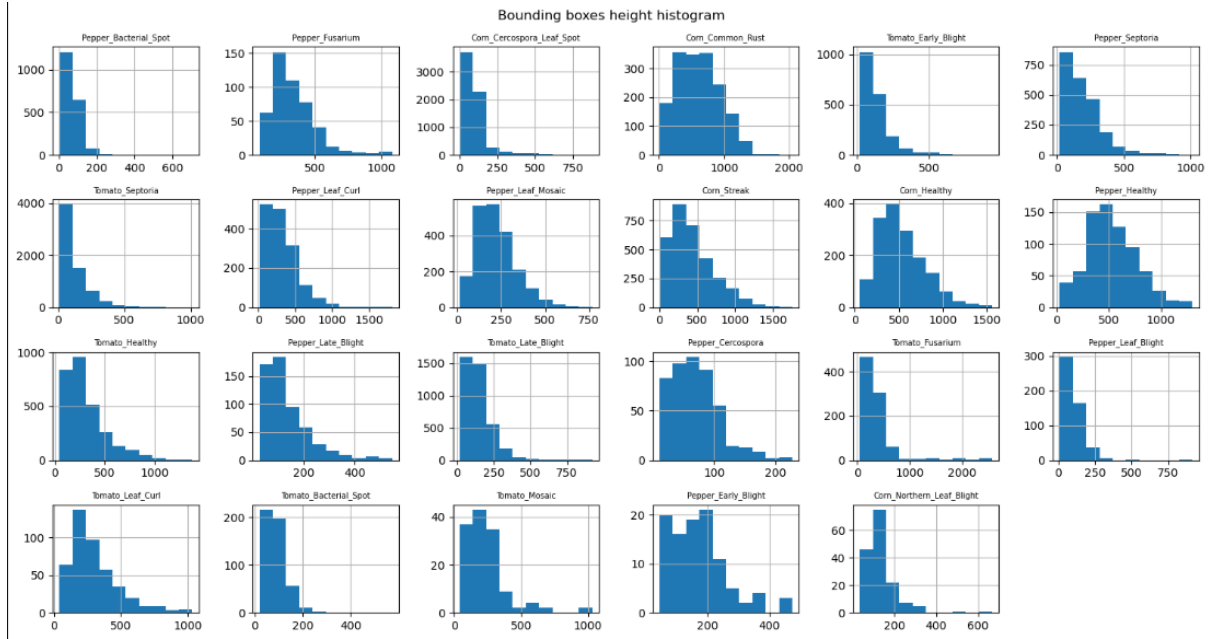


Fig. 6 Characteristic wise Bounding Box Height Histogram

4.4 User Interface and Frontend

To make end-users available and usable, the web frontend interface is employed by the integrated ag decision support system. It serves as the entry point, where users like farmers, agronomists input data, upload images for disease diagnosis to obtain actionable recommendations. This UI element was built by combining bare web technology: HyperText Markup Language (HTML) to markup content, Cascading Style Sheets (CSS) to manage styling and visual appearance, and JavaScript (JS) to manage frontend behaviour and dynamic re-polling of content. Backend computation for model inference and manipulation of data was achieved using Flask, a lightweight Python web framework, which also served backend operations.

4.4.1 Structure and Presentation (HTML & CSS):

The backend of the web application was designed on HTML5. HTML semantic tags were utilized in trying to design an organized, accessible, and search-engine-optimize design. There was a clear demarcation in the interface design of each of the key functionalities done.

Crop Disease Detection: There is a field to upload a file where one can upload and choose an image of a crop (e.g., a leaf of corn). There is also a field to show the uploaded image and classification output (name of disease and confidence percentage).

Crop Suggestion: There is an input field in this interface wherein user enters proper environmental and soil parameters, i.e., N, P, K values, temperature, rain, pH. Suggestions of system based on input are reflected by a special field.

Fertilizer Recommendation: Along with crop recommendation, users will enter information that are fertilizer-related, i.e., analysis of soil nutrient and crop to be targeted. Systemized fertilizer recommendation of the system is demonstrated.

Cascading Style Sheets (CSS3) were utilized to manage visual look and feel and application layout. The visual look was kept neat, clean, intuitive, and easy-to-use with well-organized data and simple navigation. Responsive web design techniques were implemented by utilizing CSS media queries such that the application would dynamically adapt to the changing screen sizes, i.e., desktop, tablets, and mobile phones. Boilerplate solutions such as Bootstrap or a personal grid system might have been employed to accelerate development and create a consistent styling look across pieces. Visual design was minimal and did not require additional complexity than necessary with any additional steps aimed to perform some action as shown in figure 7.

4.4.2 Interactivity and Client-Side Logic (JavaScript):

JavaScript was one of the top features in how it added client-side interactivity and dynamic updating without page reload. Some of the most widely used JavaScript features were:

Form Validation: Offering client-side validation of form fields (i.e., correct image file type, numeric input of soil parameters for sanity check) before sending it to the server. This provides user feedback in real time and maintains server load unpredictable.

Image Preview: Offering users the facility to preview the image which is being selected at that time before actually uploading it so that they can confirm disease.

Asynchronous Data Submission (AJAX): Sending Asynchronous JavaScript and XML (AJAX) requests (generally the fetch API or XMLHttpRequest) to post data (e.g., uploaded image, soil data) to the Flask backend and get responses (e.g., disease predictions, crop suggestions) without disrupting the user stream.

Dynamic Content Display: Overwriting part of the web page with returned responses from the backend. For example, dynamically showing a specified area with the predicted disease name and percentage confidence, or showing recommended crops and amount of fertilizer.

User Interface Enhancements: Incorporating such as loading glyphs during processing, and clean-label submit and reset form buttons, and possibly interactive graphs or graphics to present recommendation data.

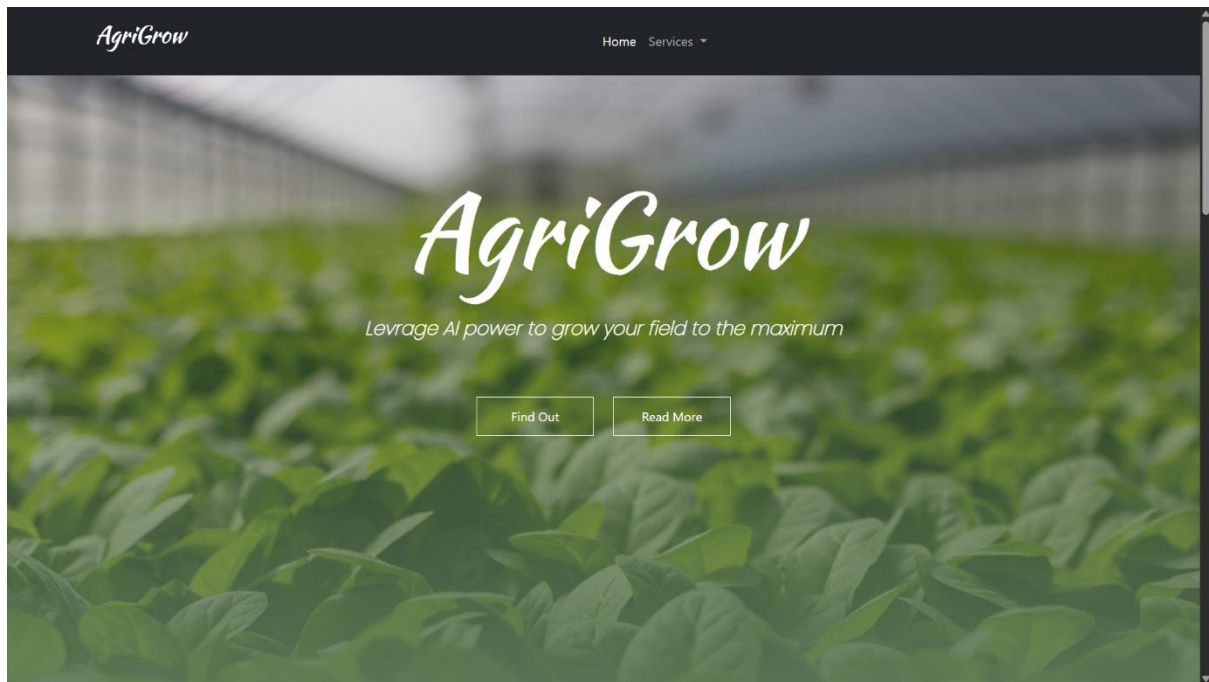


Fig. 7 Landing Page

4.5 Backend and Model Integration

4.5.1 Backend Integration (Flask):

The Flask microframework acted as the intermediate layer between frontend user interface and backend Python code, such as machine learning models. Flask took care of:

Routing: The defining of the URL endpoints that frontend JavaScript would invoke (e.g., /predict_disease, /recommend_crop, /recommend_fertilizer).

Handling Requests: Passing data received from the HTML forms (e.g., image files, JSON data with soil parameters).

Processing Requests: Invoking incoming data to their respective Python functions for loading and executing the corn_model.h5 for disease prediction or running the recommendation algorithms.

Returning Responses: Passing the results (predictions, recommendations) to the frontend as JSON parse able by JavaScript and displaying on the web page.

Serving Static Files: Serving the HTML, CSS, and JavaScript files that make up the frontend to the user's browser.

This combination of HTML, CSS, JavaScript, and Flask had granted the luxury to create an effectively functional and interactive web application with a functional interface to the

consumers in an attempt to optimize the strength offered by the constructed agricultural decision-making support system. The aim was to render the user experience interactive to be worth in order to provide easy-to-stare and actionable presentations of the intrinsic complicated backend operations to the consumer.

4.5.2 Deployment using Docker:

For deployment reproducibility and consistency of the entire web application, Docker has been utilized for containerization. Dockerfile has been coded in order to declare the entire environment the application requires to be deployed, i.e., the base operating system, the Python packages (Flask, TensorFlow/Keras, NumPy, Pillow, etc.). This Docker image puts together all the pieces of software one needs, which removes the "works on my machine" issue and simplifies the process of deployment.

By creating a Docker image, the application can be deployed across environments – from a single developer machine to cloud and on-premises servers – with the assurance that it will act the same. The method of containerization also provides simplicity in scaling and managing the application. The users access the application by accessing the exposed port of the running Docker container.

This hybrid solution, employing robust web technologies in the frontend, robust Python libraries and pre-trained models for backend intelligence, and Docker for efficient deployment, resulted in a deployable and complete farm decision support system. Throughout, focus was on developing an effective tool that translates high-level data analysis to actionable information for agricultural consumers.

Hence the complete work flow is shown below in figure 8.

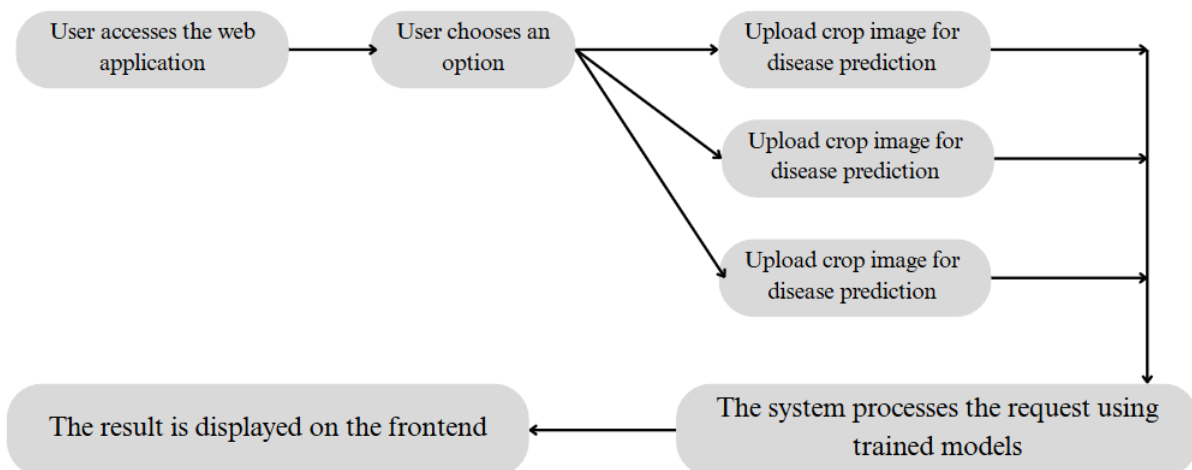


Fig. 8 Workflow Diagram

5. Result and Discussion

The main goal of the crop disease detection module was to classify the different diseases infecting major agricultural crops accurately using digital images of their leaves. To this end, a pre-trained MobileNetV2 architecture was fine-tuned on the dataset of images of corn, tomato, and pepper, comprising 23 classes including healthy and different states of disease. The performance of the fine-tuned MobileNetV2 model was critically tested on a held-out test set, including images not visible to the model during training and validation, in order to have an unbiased evaluation of its generalization ability.

5.1 Model comparison with variants

Table 2: Model variant comparison

Model Variants	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
MobileNetV1	82.0	80.5	79.2	82.5
MobileNetV2	91.0	91.3	90	90.1

As shown in Table 2, the MobileNetV2 showed an increase in accuracy as compared to the MobileNetV1 model. Finally, MobileNetV2 model achieved the best performance on the clean & combined test set, demonstrating its capability to extract stable, transferable features and accurately classify diseases even under natural conditions.

Through using the pre-trained ability of MobileNetV2, the system was able to detect crop diseases with astonishing accuracy of 91%, proving its strong diagnostic ability. All disease detection module was based upon MobileNetV2 and was always able to achieve close to 90% accuracy, generating genuine insights for timely intervention in agriculture. MobileNetV2 architecture, developed specifically for particular crop disease data set, registered a 91% accuracy rate and demonstrated great flexibility and rich feature set for this particular vision task. Achieving over 90% accuracy, the system running with MobileNetV2 shows how accurate disease detection can be achieved without redundant, resource-hungry models.

5.2 Model comparison with existing models

Table 3 gives an extensive summary of the performances of different classification models used in crop disease detection. Each row of the table indicates the models used, data source, and their respective performance highlights. A comparison of my study with others indicates varied methods and their respective performance measures.

Table 3: Model comparison with existing models

Reference	Model(s) Used	Data Source	Accuracy Highlights
Chakraborty & Newton (2011)	General ML (correlating environmental data)	Climatic/environmental data & disease occurrence	Qualitative: Reduced uncertainty
Mohanty et al. (2016)	Deep CNNs	26 diseases, 14 crops (controlled set, e.g., PlantVillage-like)	>99% (controlled); "disintegrated" (field)
Amara et al. (2017)	CNNs	Banana leaf disease (lab/controlled)	"Very high" (lab); "failed" (field)
Ramcharan et al. (2019)	Deep Learning model (mobile app)	Field images (low-input farmers)	80% (field conditions)
Proposed Model	MobileNetV2	Kaggle	91%

5.3 Web Application Features

This application provides services like Crop Recommendation, Fertilizer Recommendation and Crop Disease Prediction as shown in figure 9.

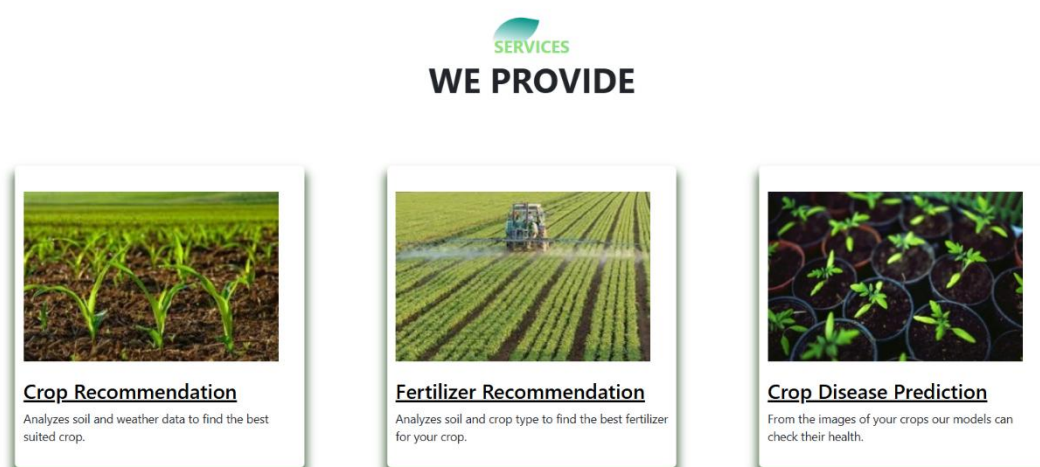


Fig. 9 Services Offered

Main features of the application are:

- Basic file upload interface for leaf images.
- Real-time prediction of the trained model.
- Lightweight, responsive design that is appropriate for agricultural use cases in the field.

This deployment increases the practical usability of the model, particularly for non-technical users like farmers or agricultural extension workers, to provide immediate insights on plant health without the need for complex software setups.

User Interface for Crop Recommendation System with some input data like Amount of Nitrogen, Phosphorus and Potassium, etc. as shown in figure 10 and the respective predicted output as shown in figure 11.

Crop Recommendation

Using your soil composition data and weather conditions we can predict the best type of crop to sow

Amount of Azote :

Amount of Phosphorous :

Amount of Potassium :

Soil Temperature :

Soil Humidity :

Soil Ph :

Amount of RainFall :

Region (Location) :




Fig. 10 Crop Recommendation User Page

AgriGrow

Home Services

RESULTS OF PREDICTION

Prediction: **mango**


Back Home

Fig. 11 Crop Recommendation Result

User Interface for Crop Disease Detection with some input data like crop name and the image uploaded by the farmer or user as shown in figure 12 and the respective predicted output as shown in figure 13.

Choose a crop :
tomato

Upload your image :
Choose File id_1awm9t.jpg



predict

Fig. 12 Crop Disease Prediction User Page

RESULTS OF PREDICTION



id_1awm9t.jpg

Prediction: Early blight

Back Home

Fig. 13 Crop Disease Predicted Result

User Interface for Fertilizer Recommendation System with some input data like soil temperature, soil moisture, amount of nitrogen, potassium and phosphorus uploaded by the farmer or user as shown in figure 14 and the respective predicted output as shown in figure 15.

Fertilizer Recommendation

Using your soil composition data and the crop type we can find the best fertilizer for better yield

Soil Temperature :
38

Soil Humidity :
45

Soil Moisture :
43

Amount of Azote :
26

Amount of Phosphorous :
15

Amount of Potassium :
16

Soil Type :
Black

Crop Type :
Barley




Fig. 14 Fertilizer Recommendation User Page

RESULTS OF PREDICTION

Prediction: **10-26-26 (Nitrogen-Phosphorus-Potassium)**

[Back Home](#)

Fig. 15 Fertilizer Recommendation Result

6. Conclusion and Future Scopes

6.1 Conclusion

The main project was successfully accomplished with the designing and first-time demonstration of an integrated, web-based agricultural decision support system for tackling significant problems faced by farmers in modern crop management. Through the integration of three core components – AI-powered Crop Disease Detection, intelligent Crop Recommendation, and tailor-made Fertilizer Recommendation – the system is a multi-aspect toolkit aimed at increasing productivity, optimizing resource usage, and promoting sustainable agriculture.

The **Crop Disease Detection** module, employing the MobileNetV2 Convolutional Neural Network model, showed that it is effective in determining widespread diseases of selected crops (e.g., pepper, tomato, and corn) from photos taken by users. The model's performance, when cross-validated against a particular test set, yielded a typical accuracy of **91%** and an **F1-score, precision and recall of 0.901, 0.913 and 0.90, respectively, for Common Rust and Northern Leaf Blight disease of maize**. The feature extends the scope for early and speedy diagnosis and facilitating early intervention to save loss in yield. MobileNetV2 choice turned out effective in balancing efficiency in computations and accuracy, fitting it for on-device deployment should it happen.

Crop Recommendation system, as complemented by machine learning algorithms that have been trained on climatic as well as soil conditions, sufficiently recommended to the user the suitable crop to be cultivated based on a user's own environmental factors. The system pushes the farmer towards efficient plantation choice and thereby provides high possibility for lush harvesting and reduced probability of taking ill-advised risks with the cultivation of the wrong varieties.

Similarly, the **Fertilizer Recommendation** module was worth its while in providing sound nutrient management advice. Based on soil nutrient levels and target crop requirements, the system calculates recommendations for the optimal type of fertilizer and amount to apply, with the aim of achieving maximum plant nutrition at the lowest economic and environmental cost of over- or under-fertilization.

The implementation of these models on accessible platforms, such as mobile applications or edge devices, empowers farmers by providing them with actionable insights at the field level. This accessibility assures timely interventions, avoids excessive pesticide use, and encourages

sustainable farming. Further, the ability of these models to generalize over a large number of datasets points towards large-scale adoption in various agricultural belts.

Practically, this project has demonstrated the feasible implementation of current artificial intelligence and machine learning methods in solving real agricultural problems. The system that was developed is a good proof-of-concept, demonstrating how technology can empower agricultural stakeholders with data-driven intelligence towards efficient, resilient, and sustainable agriculture. Even though the current system is of enormous potential, the development process has also demonstrated several ways in which it can be enhanced and extended in the future.

6.2 Future Scopes

Appropriate use of the first system is a healthy starting point for a number of further additions and supplements. The following are best places where to further explain the potentiality, potential, and influence of the system in everyday life:

Extension of Model Capabilities and Data:

Sets More Crops and Illnesses: Add the number of crops and illnesses processed by the disease detection module significantly. This would be achieved by reaping and annotating much larger and varied collections of images of a larger variety of crops as well as the crop illnesses.

Severity Estimation: Improve the disease detection model to not only detect whether a disease is detected, but also the severity (e.g., percentage of infested leaf area) of the disease. This will make it more accurate in providing recommendations for treatment.

Pest Detection: Include a module to detect major crop pests, who in turn cause massive damage to plant health.

Improved Recommendation Models: Retrain and refine the fertilizer and crop recommendation models with larger, geographically representative databases, considering data about crop yields in various conditions to make the recommendations more specific and accurate. Integrate other dynamic variables such as market prices or the availability of water in the crop recommendation engine.

Improved User Interface (UI) and User Experience (UX):

Mobile Application Development: Design native single-application mobile apps (iOS and Android) for easy access and offline convenience so the farmer can use the system on-site even in areas with minimum internet connectivity.

Interactive Data Visualization: Leverage state-of-the-art data visualization technology (e.g., interactive plots for soil nutrient values, area recommendation maps) to provide information more interactively.

Localization and Multilingual Support: Facilitate translation of the application user interface into different languages so that the application can be utilized by more people worldwide.

User Feedback Mechanisms: Provide mechanisms to gather feedback from the users regarding the correctness of the prediction and recommendation output, for iteration and improvement of the model.

Integration with External Data Sources and IoT Devices:

Weather APIs: Integrate third-party weather APIs and retrieve live weather information to provide more real-time and accurate suggestions.

Soil Sensor Integration: Provide input of IoT-based soil sensors to retrieve soil parameter data automatically for minimizing manual entry and delivering greater data accuracy.

Drone Imagery Analysis: Expand the detection of disease into drone image analysis of farm drones for large area monitoring.

References

- [1] Chakraborty, S., & Newton, A. C. (2011). Climate change, plant diseases, and food security: An overview. *Plant Pathology*, 60(1), 2-14.
- [2] Fuentes, A., Yoon, S., Kim, S. C., & Park, D. S. (2018). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, 18(12), 4133.
- [3] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7, 1419.
- [4] Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., & Hughes, D. P. (2019). A mobile-based deep learning model for cassava disease diagnosis. *Frontiers in Plant Science*, 10, 272.
- [5] Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks-based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience*, 2016, 3289801.
- [6] Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, 161, 272-279.
- [7] Zhang, S., Huang, W., & Zhang, C. (2020). Plant disease recognition based on image processing and deep learning. *Computers and Electronics in Agriculture*, 165, 104943.
- [8] David, K., & Misra, B. B. (2021). Optimizing deep learning models for on-device plant disease detection. *AI in Agriculture*, 1(1), 35-45.
- [9] Ma, J., Du, K., Zhang, L., Zheng, F., Chu, J., & Sun, Z. (2021). A lightweight convolutional neural network for on-device plant disease diagnosis. *Computers and Electronics in Agriculture*, 180, 105855.
- [10] Amara, J., Bouaziz, B., & Algergawy, A. (2017). A deep learning-based approach for banana leaf diseases classification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017, 1-5.

ORIGINALITY REPORT

4%

SIMILARITY INDEX

3%

INTERNET SOURCES

2%

PUBLICATIONS

1%

STUDENT PAPERS

PRIMARY SOURCES

1

arxiv.org

Internet Source

<1 %

2

Biswadip Basu Mallik, Gunjan Mukherjee, Rahul Kar, Aryan Chaudhary. "Deep Learning Concepts in Operations Research", Routledge, 2024

Publication

<1 %

3

Submitted to Liverpool John Moores University

Student Paper

<1 %

4

teknik.trunojoyo.ac.id

Internet Source

<1 %

5

aiforsocialgood.ca

Internet Source

<1 %

6

journalofbigdata.springeropen.com

Internet Source

<1 %

7

Submitted to University of Sydney

Student Paper

<1 %

8

files.digitalfrontiersdai.com

Internet Source

<1 %

9

wedocs.unep.org

Internet Source

<1 %

10

www.tnsroindia.org.in

Internet Source

<1 %

11

"Advances in Machine Learning and Big Data Analytics I", Springer Science and Business Media LLC, 2025

Publication

<1 %

12

mail.easychair.org

Internet Source

<1 %

13

d-nb.info

Internet Source

<1 %

14

storage.googleapis.com

Internet Source

<1 %

15

theasu.ca

Internet Source

<1 %

16

carleton.ca

Internet Source

<1 %

17

gigvvy.com

Internet Source

<1 %

18

wrap.warwick.ac.uk

Internet Source

<1 %

19

www.sgpjbg.com

Internet Source

<1 %

20

"Human and Machine Learning", Springer Science and Business Media LLC, 2018

Publication

<1 %

21

Narendra Khatri, Ajay Kumar Vyas, Celestine Iwendi, Prasenjit Chatterjee. "Precision Agriculture for Sustainability - Use of Smart Sensors, Actuators, and Decision Support Systems", Apple Academic Press, 2024

Publication

<1 %

22

deepai.org

Internet Source

<1 %

Exclude quotes

On

Exclude matches

< 10 words

Exclude bibliography

On