PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

Ministry of Higher Education and Scientific Research

Mohamed Khider University - BISKRA

Faculty of Exact Sciences, Natural and Life Sciences

Computer Science department

Order number : A_StartUp 09

Report

Presented to obtain the Master of Science degree in Computer Science

Option :Artificial Intelligence

Realized by :

-Mammeri Jamila

Title :

Predict Plant Diseases and Crop Health Analysis Using IA and IOT

Dalila Hattab	**MAA**	**Supervisor**
Tibermacine Ahmed	**MCB**	**President**
Torki Fatima Zohra	**MAA**	**Examinator**

22 June 2023

Deducation

At the end of this research, which was done with the grace and success of God, I dedicate the highest expressions of respect and appreciation to whom words cannot give her right to the candle that illuminates my life and the secret of my happiness, success, and success To the one who taught me and planted in the love of knowledge To the one who supported me and planted in the love of knowledge To the one who supported me and advised me even though she graduated from the school of life To the most precious person in existence, my mother, then my mother, then my mother Hafsiya Meghit, to the owner of prestige And reverence to the one from whom I learned to give without waiting, to the one whose nickname I carry with pride, to my dear father Mohammed Mammeri, to my guardian angel and the one who gives advice, to the well of my secrets, to the well from which I drink doses of optimism and hope every day, my beloved sister Yamina. To the feisty, affectionate, sincere, to the one who has the credit for raising the pen, to the one who taught me to study and planted in her love my beloved sister,Karima.I ask God to protect them for me, bless them for me, and protect every dear person at the end of my journey, which will be crowned with success, God willing. My brothers Ghani,Mabrouk , Rabia, Lakhdar, Sami, and Akram. The husbands of my brothers, Nariman, Nour al-Huda, Khadija, Donia, Karima, and Warda Taher, Nour al-Din. And the little smurfs pure hearts and innocent faces My nephews and sisters Alaa Hebat Al-Rahman, Mohammed, Kawthar, Muntaser Billah, Nizar Ismail, Islam, Raseem, Elias, Reem, Saleh, Ibrahim Al-Khalil, Israa, Mahdi Adam, Younes To those who helped me and with whom I tasted the sweet and bitter days Amal Saudi To my friends Najat Imran Abla bin Shawiya Maria Aqoun Sarra bin Zayadi. To the kind supervisor, Dalila Hattab, who supported me psychologically.

Jamila Mammeri

THANKS and gratitude

Praise be to God who has enabled me to do this, and I would not have reached this conciliation had it not been for the grace of the Merciful, the Merciful, who commanded him between the Kaf and the Nun, saying to him, Be, and God is the One, the One, the One Who is free from error, so praise and thanks be to the innocent.

To the one who, with her knowledge, illuminated the mind of others and showed with her humility her personality, my honorable professor Hattab Dalila, who kindly supervised this graduation Note, so I ask God to reward her with all the best, and she has all the respect and appreciation.

I also extend expressions of respect, appreciation, and many thanks to the professors of the University of Mohamed Kheidar Biskra, and I mention the professors of the Department of Computer Information, whose likes I have never seen, despite their knowledge of a lot, but it increases their humility. They respect the student. They treat us as if we were their colleagues or brothers. From near or far, I also do not forget to thank the employees of the Information Department whenever we needed them for something, we found them present.

Jamila Mammeri

Table of Contents

Deducation **i**

THANKS and gratitude **ii**

Table of Contents **iii**

List of Figures **x**

List of Tables **xv**

Introduction **1**

1 Phase-1- Plants Diseases **4**

1.1 Introduction 5

1.2 Plant Pathology or Phytopathology 6

1.3 Plant disease 6

1.4 Symptoms[1] 7

1.4.1 The most important symptoms of plant diseases[1] 7

1.5 The aim of studying plant pathology[2] 12

1.6 The history of plant diseases[2]. 13

1.7 The significance of plant diseases[2]. 15

1.8 The dissemination of plant pathogens[2]. 15

1.9 Monitoring, examination, diagnosis and prediction[1]. 16

1.9.1 Monitoring 16
1.9.2 Examination 16
1.9.3 Diagnosis 16
1.9.4 Prediction plant disease 18
1.10 Control of plant diseases[1]. 18
1.10.1 Legislative and legal methods 18
1.10.2 Agricultural methods 18
1.10.3 Vitality methods 19
1.10.4 Physical methods 19
1.10.5 Chemical methods 19
1.11 Agricultural pests[1]. 20
1.11.1 Insects 20
1.11.2 Mites (spiders) 21
1.11.3 Caecilians (Nematoda) 22
1.11.4 Flowering plants (dodder) 22
1.11.5 Weeds and wild herbs 22
1.11.6 Rodents 23
1.11.7 Microorganisms 24
1.12 Classification of plant diseases[1] 24
1.12.1 Fungal diseases 25
1.12.2 Bacterial diseases 28
1.12.3 Viral diseases 29
1.13 Agricultural disturbances[1]. 30
1.13.1 The effect of environmental factors on the plant in disease events . 30
1.13.2 The most important environmental factors affecting plant growth and development 31
1.13.3 The most important physiological diseases affecting vegetable crops 33

1.14 The disease severity . . . 37
1.15 Disease monitoring[21] . . . 38
1.16 Percentage Scales[23] . . . 38
1.17 Computer Simulation[25] . . . 39
1.18 Conclusion . . . 40

2 PHASE-2- AI and IoT in Plant Diseases 42

2.1 Introduction . . . 43
2.2 The objective of this work . . . 44
2.3 The related work . . . 45
2.4 Artificial intelligence (AI)[40] . . . 47
2.5 Machine learning (ML)[41] . . . 48
2.5.1 Types of Systems of Machine Learning . . . 48
2.6 Machine Learning Models (ML)[12] . . . 50
2.6.1 Support Vector Machine (SVM)[26] . . . 50
2.6.2 Naive Bayes . . . 51
2.6.3 K-Nearest Neighbor (KNN) . . . 52
2.7 Deep learning (DL) . . . 52
2.7.1 Batch Normalization[43] . . . 53
2.8 Deep Learning models (DL) . . . 54
2.8.1 Artificial Neural Networks (ANN)[12] . . . 54
2.8.2 Convolutional Neural Networks (CNN) . . . 55
2.8.3 Convolutional Neural Network Architectures[12] . . . 59
2.8.4 Comparison of various classifiers . . . 61
2.8.5 DCNN[49] . . . 61
2.8.6 The Difference between machine learning and deep learning[54] . . . 62
2.8.7 Image recognition technology based on deep learning[47] . . . 62
2.8.8 Open source tools for deep learning . . . 62

2.9 Transfer Learning (TL)[12] 63
2.10 Data Operations 63
2.10.1 Data acquisition 63
2.10.2 Data Pre-processing[12] 64
2.10.3 Data Analysis[34]. 67
2.11 Network type[47]. 68
2.11.1 Classification network [47]. 68
2.11.2 Detection network[47]. 71
2.11.3 Segmentation network[47] 72
2.12 Evaluation indices[47]. 73
2.13 Challenges[47]. 73
2.13.1 Small dataset size problem 73
2.13.2 Detection performance under the influence of illumination and occlusion 74
2.14 Computer vision[72] 75
2.15 Precision agriculture[73] 75
2.15.1 The benefits of precision agriculture[73] 75
2.15.2 Are there any challenges involved with precision agriculture ?[73] ? . 76
2.15.3 The use cases for precision agriculture[73] 76
2.16 Smart Farming (SF)[74] 77
2.17 Internet of Things (IoT) 77
2.17.1 Major components of IoT based Smart Farming[78] 78
2.17.2 IoT Agriculture network architecture[78] 79
2.17.3 The benefits of IoT[83] 81
2.17.4 Disadvantages of IoT[83] 82
2.18 Smart Agriculture Monitoring (SAM) 82
2.19 Overview of Wireless Communication Technologies 83

2.19.1 Wireless sensors (WS)[8] 85
2.20 Cloud Computing [98] 86
2.20.1 Cloud computing categories[98] 86
2.20.2 Deployment Models[98] 87
2.20.3 Advantages of Cloud Computing[98] 87
2.20.4 Disadvantages of the Cloud [98] 88
2.21 AGRICULTURAL DRONES 88
2.22 Conclusion 88

3 PHASE-3- REALIZATION AND IMPLEMENTATION 90

3.1 Introduction 91
3.2 Description of the prototype 92
3.3 Conception of the system 93
3.4 Materials and methods 94
3.5 How systems work ? 96
3.6 Development of Experimental Setup[86] 98
3.6.1 Field Sensors 98
3.6.2 Microcontroller 98
3.6.3 Wireless Communication Technologies and Protocols 98
3.6.4 Field Actuators 98
3.7 Description Architecture IoT used[86] 99
3.7.1 The transmitter section (TX) 99
3.7.2 The receiver section (RX) 99
3.7.3 Field Estimation for Deployment Sensors 101
3.8 Applications of the proposed methodology 102
3.8.1 Plant and crop management 102
3.8.2 Crop disease and pest management 103
3.8.3 Crop recommendation[106] 103

3.8.4 Soil monitoring[8] . . . 104
3.8.5 Fertilizer system . . . 104
3.8.6 Irrigation system . . . 105
3.8.7 Weed detection. . . . 106
3.8.8 Advantages of the system[110] . . . 106
3.9 IoT sensors . . . 106
3.9.1 NPK sensor . . . 106
3.9.2 The DHT . . . 107
3.9.3 Soil Moisture Sensor[35] . . . 107
3.9.4 Rain sensor[114] . . . 108
3.9.5 Raspberry Pi[115] . . . 108
3.9.6 NODEMCU[35] . . . 110
3.10 Software and Libraries used . . . 111
3.10.1 Python [110] . . . 111
3.10.2 TensorFlow[35] . . . 113
3.10.3 PyTorch . . . 113
3.10.4 Scikit-learn . . . 114
3.10.5 Keras . . . 114
3.10.6 OpenCV[79] . . . 114
3.11 Platforms Used . . . 115
3.11.1 Google Colab Notebook . . . 115
3.11.2 Kaggle Notebook . . . 115
3.12 Algorithm prediction of plant disease using sensor and camera . . . 116
3.13 Training Models [35] . . . 117
3.14 Dataset Description . . . 118
3.14.1 Dataset preparation . . . 119
3.14.2 Problematic situations and indicative cases[77] . . . 120

3.15 Analysis and Design 120
3.15.1 System Architecture 120
3.15.2 Use Case Diagram 121
3.15.3 Sequence Diagram 122
3.16 Implementation 123
3.16.1 Results and Discussion 123
3.16.2 Plagiarism 172
3.17 Conclusion 172
3.18 Conclusion : 175

Bibliographie **177**

Annexe A : Abbreviations and notation **190**

List of Figures

1.1 Diagram showing the basic functions of plants on the left and plant diseases on the right. [2] 7

1.2 Some key inputs, processes involved and possible outputs of smart farming [8]. 13

1.3 Major challenges for sustainable future agriculture [8]. 14

1.4 The world population from 1950 to 2020 [9]. 14

1.5 A picture of the white fly and Aphids or Manna. [1]. 21

1.6 Cutworms (cutting worms are characterized by their sudden appearance in scattered spots in the field). [1]. 21

1.7 The red spider (infects all year round vegetable crops, especially the cucurbit family, of which cucumber is one) [1] 21

1.8 Nematodes (includes within the circle of parasitism and Root-knot nematode disease [1] 22

1.9 Halouk (Jafail, lion lentil, chokeberry, devil's clover, rabbit bread, a parasitic flowering plant belonging to the Jafail family and attacks nightshade, legumes, and many other crops) [1] 23

1.10 Al-Khaland (the animal was given the name (Abu Amaya) (because it has closed eyes and does not see, It spends most of its life underground, digging tunnels and forked, shallow ground burrows between the roots, bulbs, and tubers of the plants that it feeds on and stores, and thus causes them to perish . [1] 23
1.11 A chart showing the division of plant diseases according to the pathogen. . 25
1.12 Seedling fall disease and root rot. [1] 26
1.13 Early blight (symptoms appear in the form of small dark brown spots that soon turn black, especially on the old lower leaves). [1] 26
1.14 Late blight (its color is initially dark brown on the upper surface of the leaf, and there are white fluffy fungal growths on the lower surface, then it turns black, which leads to drying and falling of the leaves, as well as distortion of the fruits) See [1] 27
1.15 Anthracnose [1]. 27
1.16 Bacterial mold (the first thing that indicates the presence of the disease is the appearance of soft watery spots, which grow with time, and increase in depth in the affected plant tissue, until all infected parts become soft at the end, and the color of the outer surface of the infected part changes and most parts of the plant that are exposed to this disease They are fruits or storage organs such as tubers. [1]. 28
1.17 Bacterial spotting. [1]. 29
1.18 Tomato mosaic. [1]. 29
1.19 Watermelon mosaic. [1]. 30
1.20 Image illustrating the relationship between the pathogen, host and vector in the presence of environmental conditions [2]. 31
1.21 This table presents some numerical Range of features for a particular disease. 32
1.22 Circulacrack , transverse crack and recessed crack [1]. 33

1.23 Sun blight. [1]. 33
1.24 The central gaps in the seeds . [1]. 34
1.25 Irregular fruit discoloration or spots. [1]. 34
1.26 Nutrients and their role. 35
1.27 Places of appearance of symptoms of nutrient deficiency on the plant . [20] 37

2.1 Supervised learning example. [41] 48
2.2 Unsupervised learning example . [41] 49
2.3 Reinforcement learning example [41] 50
2.4 Support Vector Machine. [42] 51
2.5 Summary of the deep learning evolution from 1943 to 2021 See [49] 53
2.6 ANN example [50] 54
2.7 Convolution Layer [35] 56
2.8 Relu Layer [35] 56
2.9 The process of pooling operation. [52] 57
2.10 Fully connected layer. [35] 58
2.11 A typical CNN architecture for plant disease classification [52] 58
2.12 Comparison of various classifiers [53] 61
2.13 Contrast between traditional image processing and deep learning methods. [47] 62
2.14 Comparison of open source tools for deep learning [47] 63
2.15 An example of background removal [16]. 65
2.16 Definition of plant disease. [47]. 68
2.17 Comparison of each sub-method of classification network [47]. 69
2.18 Important measures in a statistical classification task [101]. 73
2.19 Sample illustration of smart farming. [75] 77
2.20 Major components of IoT based smart farming [78] 79
2.21 Overview of the smart agriculture parameters. [85] 83

2.22 Comparison of existing wireless protocols [78] 84
2.23 Various wireless communication technologies for range and data rate comparisons [86] 84

3.1 General conception of my system. 93
3.2 Architecture represents how system work. [98] 94
3.3 Architecture of Hardwar component of AI- AgriPlantD. 94
3.4 General architecture of AI- AgriPlantD. 95
3.5 Proposed architecture of AI- AgriPlantD with IoT. 96
3.6 Architecture of operation of prediction disease by image. 97
3.7 Proposed architecture for the smart agriculture decision-making system [86] 101
3.8 Packet loss rate versus distance [86] 102
3.9 Diagram of classification plant disease [56]. 103
3.10 Diagram of recommended crop for irrigation. [106] 104
3.11 Schematic diagram of irrigation system. [104] 106
3.12 Comparison between Specifications and features of DHT11 and DHT22(or AM2302) sensor [113]. 107
3.13 Soil moisture Sensor. 108
3.14 Rain sensor [114]. 108
3.15 Raspberry Pi [116]. 109
3.16 Arduino Vs Raspberry Pi . [117] 110
3.17 WiFi module [114]. 111
3.18 Algorithm of prediction plant disease. 116
3.19 How information pass in sensor. [106] 117
3.20 Proposed IoT Client-server model. [106] 118
3.21 Example of leaf images from the PlantVillage dataset, representing every crop-disease pair used . [108] 118
3.22 Architecture of AgriPlantD application. [120] 121

3.23 UML Diagram – I 121
3.24 UML Diagram of predict plant disease by image – II 122
3.25 UML Diagram of predict plant disease by sensor data – III 122
3.26 Code source first model. 124
3.27 Comarative study between my models and previous models 141

List of Tables

1.1 Some nutrients and their role 36

Introduction

The implementation of suitable control techniques, such as the use of insecticides, fungicides, and disease-specific chemicals, which eventually result in increased production, depends critically on the effective monitoring of crop health and early disease identification. Plant illnesses brought on by bacterial, fungal, viral, and insect infestations can cause major harm and have a negative influence on agriculture. Early disease detection can reduce losses and stop the spread of the illness.

Particularly in nations like Algeria, where it employs about 14.7% of the population, agriculture is a crucial sector. However, choosing the right crop for a farmer's plot of land can be difficult, which reduces yields and earnings. There is an urgent need to expand food production since the world's population is expected to exceed 9.7 billion people by 2050. A 70% increase in food production is urgently needed. Pests and diseases, on the other hand, constitute a danger to agricultural output, leading to considerable losses and even crop collapse. Precision farming can be revolutionized by utilizing technology like the Internet of Things (IoT) and image processing for disease prediction. This will allow for the effective monitoring of enormous agricultural fields and the precise detection of plant diseases using deep neural networks.

Large farms cannot be monitored using the traditional disease detection approach, which relies on human competence through visual inspection. As a result, crop monitoring and forecasting require automated techniques. By using such solutions, you may lessen environmental harm and related production costs caused by overusing pesticides and other chemicals. Furthermore, improvements in big data analysis present prospects for smart farming by offering knowledge for increased agricultural yields, sustainable practices, disease detection, and effective management of water and soil. The gathering and analysis of big data in agriculture use information from a variety of sources, including meteorological stations, remote sensors, historical records, and publicly accessible databases.

With roughly 690 million people today experiencing food insecurity, the alarming rise in world hunger emphasizes how critical it is to correctly identify and identify plant diseases. Pests and diseases cause more than 50% of agricultural output to be lost, which has an enormous influence on the food supply. In the past, plant diseases have had disastrous effects, such as the Irish potato famine that killed millions of people. Therefore, by evaluating real-time data on air temperature, humidity, and other elements, coupled with photographs of plant leaves, advanced technologies, such as artificial intelligence, IoT, and cloud computing, can enable early prediction of plant illnesses. This information can make it easier to identify diseases, give descriptions of diseases and their symptoms, and suggest treatments.

The concept also includes a fertilizing system that selects cultivable crops depending on factors like soil type and climate. A technique for identifying weeds and insects also aids in combating pests that harm plants by consuming their sap. Farmers can optimize irrigation time and volume with the help of an intelligent irrigation system. Along with training and counseling programs, a platform is also developed to promote cooperation, knowledge exchange, and education among growers, plant breeders, specialists, and consumers.

The project's main objective is to identify early-stage plant illnesses and prevent them, reducing economic and psychological costs for both farmers and consumers. The initiative seeks to stabilize prices and guarantee that consumer living expenses are reasonable by preventing crop losses. Since all data is saved on the cloud, farmers may access it from any location at any time. The data is processed by pre-built artificial intelligence algorithms, which produce superior outcomes to earlier research. The farmer receives messages via their phone with information about the progress of the plants, as well as suggestions and instructions.

The project's succeeding sections are organized as follows : Plant diseases and agricultural pests, their sources, and their effects are thoroughly explained in Section 1. The main ideas of artificial intelligence, the Internet of Things, and other materials and procedures that are required to achieve the project's objectives are examined in Section 2. The experimental design, data, analysis, and conclusions are all included in Section 3.

Chapter 1

Phase-1- Plants Diseases

1.1 Introduction

Plant diseases pose significant challenges to agricultural productivity, affecting crop yield, quality, and overall plant health. They are caused by various pathogens, including fungi, bacteria, viruses, nematodes, and other microorganisms. Plant diseases can manifest as infections in leaves, stems, roots, or fruits, resulting in symptoms such as wilting, discoloration, lesions, stunted growth, and yield loss.

Understanding and managing plant diseases is crucial for sustaining global food production and ensuring food security. Effective disease management involves a combination of preventive measures, early detection, accurate diagnosis, and appropriate control strategies.

Plant diseases can spread through various means, including infected seeds, contaminated soil, air-borne spores, water, and insect vectors. Factors such as environmental conditions, crop variety susceptibility, and cultural practices can influence disease development and severity.

Plant pathologists and researchers study the causes, biology, and epidemiology of plant diseases to develop strategies for disease prevention and control. They conduct extensive research to identify resistant crop varieties, develop disease-resistant traits through genetic modification, and devise sustainable and integrated disease management approaches.

Disease management strategies may include cultural practices like crop rotation, sanitation, and proper plant spacing to reduce disease incidence. Chemical control methods, such as the use of fungicides and bactericides, can be employed when necessary, although their usage needs to be judicious to minimize environmental impacts and the development of resistance.

Biological control methods, utilizing beneficial organisms such as predatory insects or antagonistic microorganisms, are also employed to suppress plant diseases. Additionally, advancements in precision agriculture, remote sensing, and data analytics are being harnessed to monitor and predict disease outbreaks, aiding in timely interventions.

It is crucial to note that plant diseases can have significant economic implications, leading to significant losses for farmers and affecting global food supplies. Vigilance, early detection, and proactive disease management are vital to minimizing the impact of plant diseases on agricultural systems and ensuring sustainable food production.

In conclusion, plant diseases are a persistent threat to agriculture and global food security. Ongoing research, technological advancements, and collaborative efforts among scientists, farmers, and policymakers are necessary to develop effective disease management strategies and safeguard our crops against devastating diseases.

We will present in this section :

1 -What does plant disease mean ?

2 -Definitions of the most important concepts.

3 -What are the harms of plant disease, and the aim of studying plant disease ?

4 Some plant diseases : their causes, symptoms...-

1.2 Plant Pathology or Phytopathology

It is a science that studies plant diseases.

1.3 Plant disease

It is any abnormal deviation and continuous change that occurs during plant growth and activity arising from the presence of a pathogen, which causes the appearance of symptoms that hinder the plant, which negatively affects production in quantity and quality.[1]

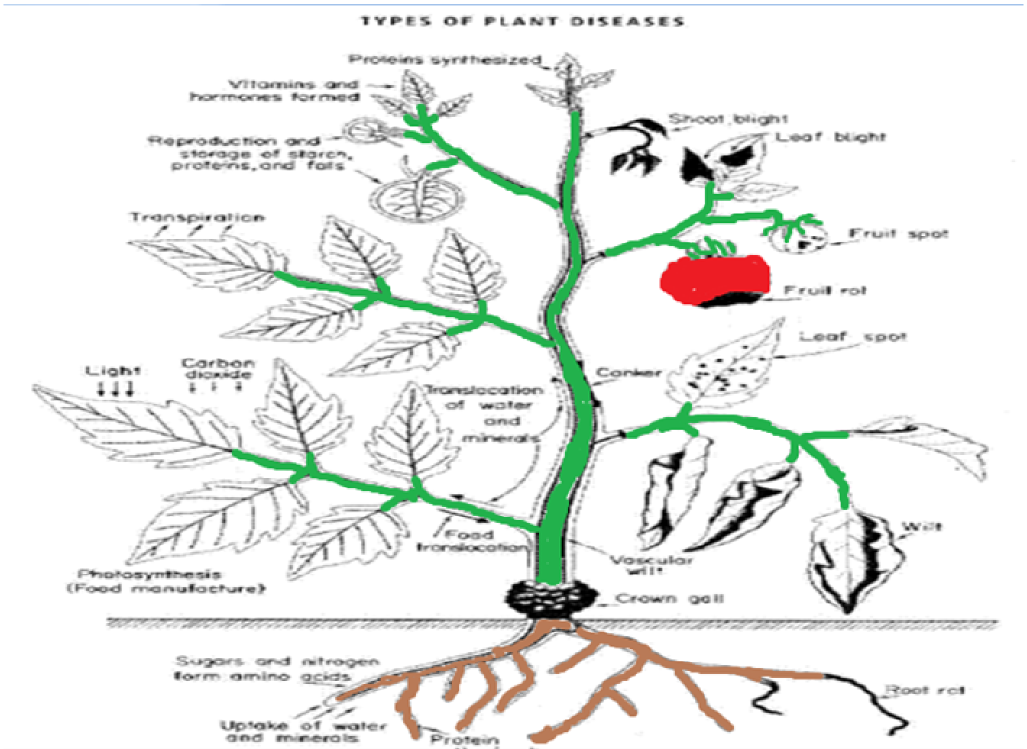


FIG. 1.1 – Diagram showing the basic functions of plants on the left and plant diseases on the right. [2]

1.4 Symptoms[1]

It is the group of changes that appear on the diseased plant, for example, the change in the color, texture, and shape of the plant tissue. Often we can do a field diagnosis of the plant through the symptoms, and if it is difficult, we do a laboratory study.

1.4.1 The most important symptoms of plant diseases[1]

The discoloration

Appears mainly on leaves, but can also appear on flowers, fruits, stems, and roots. Among its forms :

Chlorotic paleness It is the deterioration of the green pigments of the plant tissue (chlorophyll) and its coloration in light or pale green color, and it may appear sometimes in the veins only or on the blade between the veins, and it can occur due to (lack of nitrogen or iron or viral infection).

Whitening That is, the color becomes completely white, and it occurs due to the absence of all existing plant pigments, and this can be of genetic origin (which may appear on part or all of the plant) or of external origin (due to herbicides that affect chlorophyll).

Yellowing It is a decline in chlorophyll pigment followed by the appearance of a yellow pigment, resulting from carotenoids or xanthophylls. , and it can occur for physiological reasons, not innate, and may result from infection with viruses or mycoplasma.

Redness This the emergence of red pigments, either due to the deterioration of the green pigments, which allowed the emergence of red pigments (anthocyanins) that are naturally present in the tissues, or their production abnormally as a result of a pathological injury.

Mottled It is an irregular change in color, as alternating areas of varying color intensity appear, and it is considered a preliminary indication that the plant is infected with a viral disease.

Dark green Which is an increase in the intensity of the green color, which makes the vegetative organs have a bluish appearance. And its symptoms usually appear in the case of phosphorus deficiency or an increase in nitrogen in the plant concerned, or in the case of extreme thirst.

Darkening of tissues Is the accumulation of melanin, which results from a pathological action in the form of dark-colored compounds, and is often attributed to the action of oxidase enzymes on phenolic compounds.

Symptoms of wilting

It is a loosening of the juicy plant organs accompanied by yellowing or drying and falling of the leaves, then the death of the branches, and perhaps the death of the entire plant when the imbalance in its water balance worsens, and it may be sudden or gradual, and its causes may be :

1 Environmental physiology, such as dry weather and intense heat, as it is related to the scarcity of water in the soil or to the increased loss of water from the leaves.

2 Parasitic, such as rot that affects the roots or wilting fungi that enter through the root and settle in the wood and impede the access of water and mineral elements to the upper parts of the plant.

Symptoms of necrosis or localized death

It is the death of tissues and cells as a result of an inevitably parasitic invasion of fungi and bacteria. As for inevitably parasitic organisms such as viruses, rust fungi, or downy and powdery mildew, they do not destroy plant tissue, if they do not fall within the symptoms of this group. Among them are its most important forms :

Blight It is the sudden death of parts of the entire plant (the stem and branch with its flowers, leaves, and then fruits, and their coloration in a dark brown color without falling off). It results from infection with a rapidly multiplying fungus or bacteria (such as fire blight and monilia blight
on fruit trees).

Damping-off It is the curvature of the seedlings and their lying on the soil as a result of a local decomposition of the plant tissue near the soil surface.

Back Die It is the gradual death of the tips of the stem or branches and twigs starting from their tops.

Spotting The appearance of dead spots of specific shape and area on young plant organs such as leaves and fruits. The shape, diameter and color of the spot differ from one plant to another .The characteristics of the spot are An important indicator in the diagnosis Most of the blotches are caused by the intrusion of astrocytic or incomplete fungi.

Shot-hole It is a form of blotch that affects the leaves and ends with the detachment of the plant tissue and its fall leaving limited holes.

Blotch It is similar to blotch However, the dead zone is not defined in size or shape .

Scab abnormal superficial rotting of the plant cells which later leads to roughening and cracking of the surface of the fruit or tuber at the site of infection.

Russet Which is the rotting of the epidermal cells in the fruits as a result of their infection by surface parasites such as powdery mildew, their sensitivity to pesticides, or exposure to unsuitable weather conditions. The corking of straw is less deep and cracked than the corking of scabs.

Rot Usually seen on fruits, tubers and bulbs, and sometimes on roots and stems. Where the causative agent secretes enzymes,especially pectin and cellulose decomposing enzymes, which leads to the decomposition of the cell walls of the plant tissue and the exit of the cytosol. And putrefaction is often accompanied by smells of fermentation. The mold may have a distinctive color, which is the color of the mycelium of the causative fungus, so it is said, for example, blue, gray, green or brown mold.

Mummification Which is a subsequent presentation of mummification, where the rotten fruit dries up and remains attached to the plant or falls to the ground. It is called mummified or mummy. The mummified organism usually contains mold-causing structures, such as static mycelium or stony bodies.

Canker (Chancre) Which is a localized death of perennial tissues of wood and bark on the branches, stems and roots. In the case of a canker, the organism is in constant conflict with the plant tissue that tries to close the ulcer with tissues generated by the cambium adjacent to the dead tissue.

Symptoms of deformity

Nanism It is the failure of the plant or some of its members to reach their normal size. Nanism is often caused by viral, mycoplasmic and genetic diseases.

Tumor Which is an abnormal enlargement of plant tissue caused by the excessive division of cells as a result of an increase in cell sizes, or both. And it has two forms, the first occurs as a result of the secretion of substances of an oxygenic nature that incite excessive growth, as is the case in the bacterial tuberculosis of olives and the root nodes caused by nematodes. The other form is more like real cancer, where the pathogen causes a genetic change in the plant cell, turning it into a carcinogenic cell with random division, as is the case in bacterial coronary tumor disease.

Curl In which the leaves curl and twist in irregular shapes, and the curl is usually accompanied by irregular thickening of the leaf tissue, as is the case in peach and almond leaf curl disease.

Rosette Where the internodes of the stem are dwarfed, and the leaves come close to each other, and thus they can be likened to the petals of roses.

Phyllody This means the phyllody of the flower, as it is surrounded by green leaves instead of the petals. It is a common symptom in some mycoplasmic diseases, such as tomato color disease.

Root hairy Where abundantly branching and very thin roots are formed, as in bacterial hairy root disease. G - Proliferation : Which is the formation of a plant organ in an abnormal way on another similar organ, such as the formation of overlapping flowers in chrysanthemums or the formation of lateral pubes on the original cob of corn ? It is often a symptom of genetic abnormalities.

1.5 The aim of studying plant pathology[2]

1 It is to prevent the occurrence of plant epidemics by studying the behavior of pathogens, the spread of epidemics and how they occur, and then trying to break the pathological cycles in their weaknesses, as by reducing the amount of infectious inoculum, the disease can be controlled.

2 To maintain high yields and limit crop losses.

3 Ameliorate conventional agricultural systems.

4 climate change also affects plant growth by reducing water availability and other abiotic stress factors. Recent studies have shown that climatic variables (e.g. humidity, temperature) are important drivers explaining the global distribution of potentially pathogenic soil-borne microorganisms [3, 4]

5 Diseases are responsible for losses of at least 10% of global food production, representing a threat to food security (Strange & Scott, 2005) [5]. Agrios (2004) [6] estimated that annual losses by disease cost US$ 220 billion. Besides direct losses, the methods for disease control - especially the chemical methods – can result in environmental contamination and residual chemicals in food, in addition to social and economic problems. The close relationship between the environment and diseases suggests that climate change will cause modifications in the current phytosanitary scenario. The impacts can be positive, negative or neutral, since there can be a decrease, an increase or no effect on the different pathosystems, in each region.[7]

6 Advancements in technology, including sensors, communication methods, machines, and even robots [8].

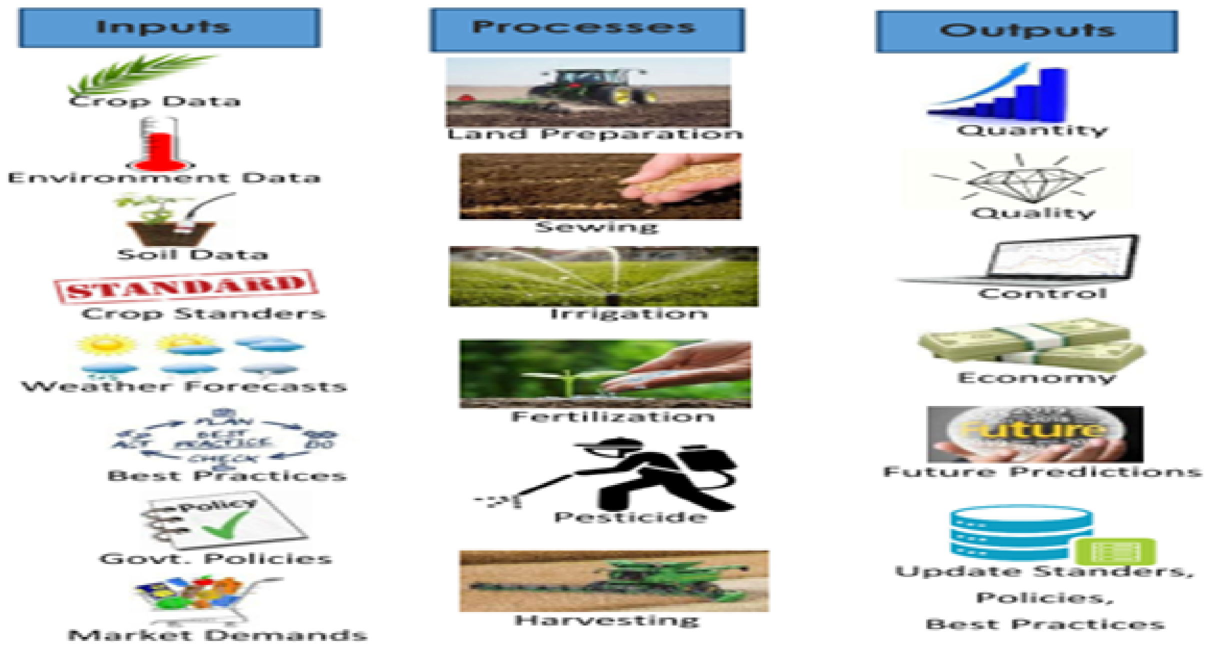


FIG. 1.2 – Some key inputs, processes involved and possible outputs of smart farming [8].

7 Economic and crop production losses.

8 The big world population.

1.6 The history of plant diseases[2].

- Man tried to resist plant diseases around 700 BC when the Romans made offerings to the rust gods, as they believed that the pathogens were evil spirits.
- Generations passed on these superstitions until they became a fixed belief near the end of the 18th century AD.
- At the end of the 18th century, the Englishman Forsyth performed the first surgery on a tree, where he got rid of the wounds and ulcers that appeared on its trunk and coated it with a paste used to treat cows' wounds, and the tree was cured.

FIG. 1.3 – Major challenges for sustainable future agriculture [8].

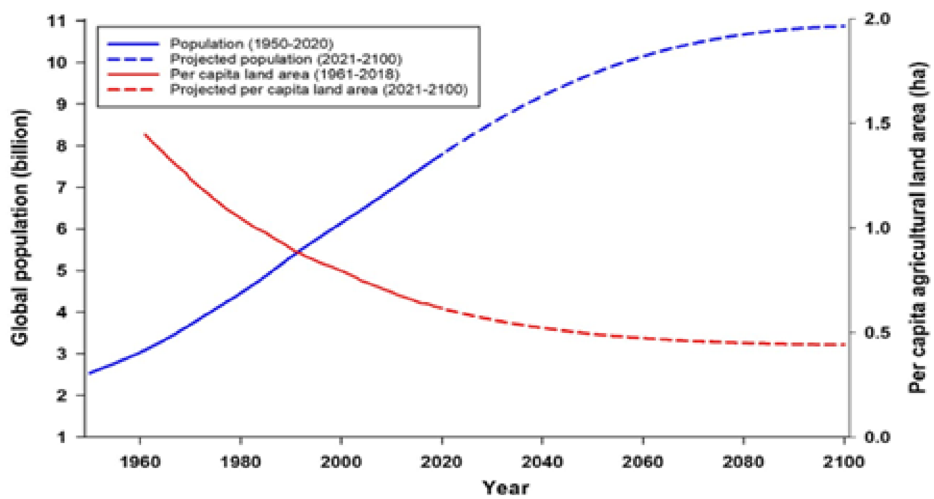


FIG. 1.4 – The world population from 1950 to 2020 [9].

- In modern times, the French proved it Anton deBary there is a fungus associated with potato blight disease, and it is the cause of the disease symptoms.
- In 1824, sulfur was used to protect plants from powdery mildew.
- In 1878, Thomas Burrill discovered the cause of fire blight in the pear, which is bacterial, and this was the first record of the occurrence of a plant disease caused by bacteria.
- During World War II and with technological development, scientists were able to manufacture many pesticides and their use spread even in the fight against bacterial

plant diseases.

- With the advent of the term artificial intelligence, scientists tried to make it easier for farmers to create smart systems to diagnose and predict plant diseases.
- With the advent of the term Internet of Things, the world has become oriented toward smart and accurate agriculture, which allowed diseases to be detected easily.

1.7 The significance of plant diseases[2].

- Its occurrence causes damage and destruction to the plants on which man depends for his food, as its deficiency leads to misery and may end in starvation, as is the case in poor countries.
- Doom, death and mass emigration, as happened in Ireland in 1845, and Japan in 1941...
- Determine agriculture and industry, as diseases control the type and quantity of industry and agriculture according to each region.
- Quantitative and qualitative loss in the crop, and some of it turned into an image unfit for human use.

1.8 The dissemination of plant pathogens[2].

Diffusion is the movement from one field to another, from one country to another, from one region to another, or from one continent to another, due to several factors :

- By wind, rain and water.
- By insects, animals, humans, soil and seedlings.
- By plant waste.

1.9 Monitoring, examination, diagnosis and prediction[1].

1.9.1 Monitoring

It is a process of follow-up, monitoring and counting to determine the number of insects present in the field environment, and it is done by using sticky color traps distributed inside the field in a regular distribution, and their number is not less than (50-100) a dunum according to the type of cultivation used (covered - uncovered) and if the average number of attached insects was more than 20 insects/trap, then a control decision could be taken.

1.9.2 Examination

It is a precautionary process aimed at determining the health of plants in the field, and it is done using fixed, selected examination areas distributed randomly within the field, which are visited weekly and their plants are examined for infectious pathogens, through which the first diseased spot is inferred within the field and then preventive measures are taken to contain the disease and confront it to prevent its spread.

1.9.3 Diagnosis

It is a deduction process through observations, discussions and information to identify the pathogen and then take the necessary control measures.

Observing a system or pattern of the distribution of the disease in the field

May contribute to some extent in determining the causative agent, as :

- The appearance of symptoms in a random manner on plants in the field due to one air- borne fungus or to one of the non-living factors in the soil, such as the lack of elements, or in the atmosphere as the effect of air pollutants .

- The appearance of the condition as scattered spots in the field is due to one of the soil borne pathogens such as root rot and vascular wilt diseases. The diagnostician should note whether there is a relationship between the distribution of the condition and the topography of the field.
- The appearance of the condition at the edge of the field is likely to be due to an insect-borne pathogen.

Note the distribution of symptoms on the plant

It varies according to the pathogen and sometimes depends on the environmental conditions in the case of a single pathogen. Some pathogens are characterized by infection occurring in young leaves, and there are pathogens that infection occurs in advanced leaves. Some pathogens may prefer the lower leaves because they need high humidity that is more available in the lower part of the plant, near the soil surface.

Examine the symptoms and signs accurately on the plant

Each type of defect indicates the type of disease, like this :

1 A deficiency in the growth of organs and tissues :

Such as cases of general dwarfism, short plant internodes, root failure, irregular formation of chlorophyll or other pigments, or flowering failure or holding fruits.

2 Death of tissues or parts of the plant :

These are probably the most common types of symptoms and include leaf mottling, edge burning, complete leaf death, terminal death of branches, and fruit rot .

3 General change in appearance :

It includes cases of irregular distribution of chlorophyll in leaves and flowers, such as mosaic patterns and discoloration of flowers .

4 The occurrence of an increase in growth from the normal state :

Such as tumors, curling leaves, knots on the stems and roots, and abundant flowering .

1.9.4 Prediction plant disease

It is the detection of early-stage plant disease, the ultimate goal of such detection systems is to identify the disease with a minimum of physical changes to the plant. Identifying diseases or abiotic problems as early as possible has obvious benefits. By using AI technology and IoT. we can realistically hope to identify stress symptoms before a human observer [10].

1.10 Control of plant diseases[1].

1.10.1 Legislative and legal methods

The process of enacting laws for the import or entry of agricultural materials such as seedlings, seeds, fruits, vegetable crops or grains is an important preventive means against the spread of plant diseases from areas affected by them to areas free of them. The state can enter by setting special controls for these methods. Achieved either through :

- Agricultural quarantine.
- Examination and inspection of plants in the fields.

1.10.2 Agricultural methods

It aims to get rid of pathological infections by controlling how different plants are grown, especially during the planting season and the crop that must be grown to avoid pathogens or their vectors. It also includes the removal of infected plants that may pose a threat to healthy plants or create unsuitable conditions for the reproduction of the diseased pathogen. The most important methods can be summarized as follows :

•Eliminating the plant host.

•Riddance the sources of infection.

•Ameliorate plant growth conditions and the creation of unsuitable conditions for the pathogen.

1.10.3 Vitality methods

It is the use of different methods by living organisms to riddance the pathogen, including :

•Resistant varieties.

•Accidental control and interference.

• Unusual parasitism and the case of antagonism.

1.10.4 Physical methods

These include the use of heat treatments in the control of plant diseases, the most important of which are :

•The use of high or low temperatures.

•The use of solar nitration.

•Use of radiation.

1.10.5 Chemical methods

This method is considered one of the most common means of control with many plant diseases, whether in the field, nursery, greenhouse, or during storage of agricultural products. The main purpose of using chemical pesticides, whether in liquid, powder, or gas form, is to prevent the diseased pathogen from establishing itself or causing infection in the plant host, and then to reduce the damage or economic loss to the plant as a result of the disease, especially with high-value crops.

Methods of using chemical control

It can be divided according to the part to be treated with a chemical pesticide as follows :

•Spraying and fogging the shoots.

•Treatment of the reproductive parts of the plant.

•Soil treatment.

•Treating tree wounds. Sterilization of agricultural crop storage areas. Control of insect vectors.

1.11 Agricultural pests[1].

It is every living organism that infects the plant with damage that makes its activity out of the ordinary, which contributes to the emergence of visible symptoms, external to the plant or internal to its tissues, that may hinder the growth and development of the plant naturally, which negatively affects production. The diseases that result from these pests are called infectious diseases, i.e. those whose symptoms begin to appear limited (spot), then the disease spreads little by little and quickly kills the plants and becomes an epidemic in the field, which can spread to neighboring fields, in light of the availability of environmental factors and conditions suitable for its spread growth and activity. These pathogenic organisms can be divided into the following :

1.11.1 Insects

They infect plants when they feed on plant sap, which harms the plant and causes damage, tingling,tearing and holes. As a result of this damage, pathological symptoms appear. Most of these insects transmit viruses from one infected plant to another, which contributes to the rapid spread of diseases.

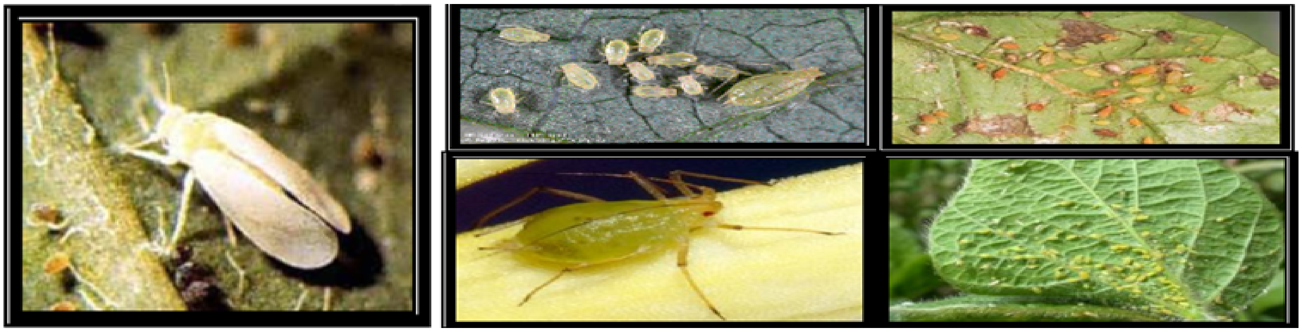


FIG. 1.5 – A picture of the white fly and Aphids or Manna. [1].

FIG. 1.6 – Cutworms (cutting worms are characterized by their sudden appearance in scattered spots in the field). [1].

1.11.2 Mites (spiders)

They are a group of animals that include (red, white and citrus rust) spiders. They cause damage when they feed on the sap of plant tissues, especially leaves, flowers, and also fruits, which causes exhaustion of the affected plant and distortion of the product.

FIG. 1.7 – The red spider (infects all year round vegetable crops, especially the cucurbit family, of which cucumber is one) [1]

1.11.3 Caecilians (Nematoda)

They belong to the animal kingdom, some of them live as saprophytes, and others are obligatory or facultative parasites, some of which cause serious plant diseases on most field crops, vegetables and fruits. Worms that parasitize the root system of plants give symptoms of weakness, yellowing, and stunting of the affected plants with the appearance of symptoms of malnutrition. However, these symptoms may be confused as many other plant pathogens cause similar symptoms. Therefore, to diagnose such cases, the root system must be examined. Where some types of caecilians show clear and distinct symptoms such as the occurrence of root knots, but in most cases a careful microscopic examination is needed to identify the true cause.

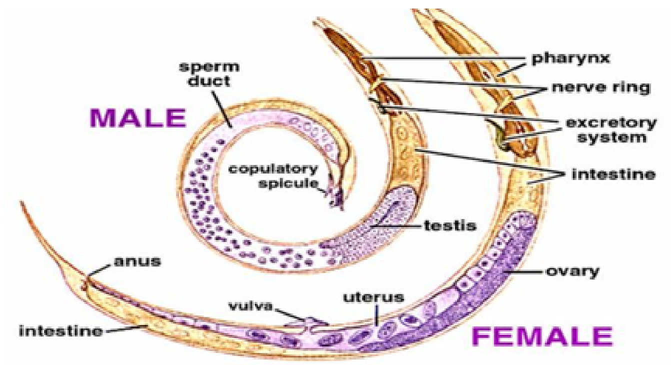


FIG. 1.8 – Nematodes (includes within the circle of parasitism and Root-knot nematode disease [1]

1.11.4 Flowering plants (dodder)

It is a plant group that lives a parasitic life on the plant and feeds on plant sap, which affects its growth, causing exhaustion and weakness.

1.11.5 Weeds and wild herbs

A plant group that grows inside the field and crowds plants by competing for sources of growth resources (food, water, soil), which hinders its growth and development, leading to

Fig. 1.9 – Halouk (Jafail, lion lentil, chokeberry, devil's clover, rabbit bread, a parasitic flowering plant belonging to the Jafail family and attacks nightshade, legumes, and many other crops) [1]

weak plant growth and a lack of crop yield, such as dodder (bean, tomato, clover) ,subway, orobanche Egyptian , Orobanche minor, eclipta alba and orobanche crenata.

1.11.6 Rodents

It is an animal group that includes mice and moles. They cause severe damage to crops, as they feed on seeds, shoots, and fruits.

Fig. 1.10 – Al-Khaland (the animal was given the name (Abu Amaya) (because it has closed eyes and does not see, It spends most of its life underground, digging tunnels and forked, shallow ground burrows between the roots, bulbs, and tubers of the plants that it feeds on and stores, and thus causes them to perish . [1]

1.11.7 Microorganisms

A group of organisms characterized by being minute, which may be seen with the naked eye, a regular microscope, or an electron microscope, and they cause many diseases to plants as a result of their parasitism on it, and they are divided into the following :

Fungi

They are microorganisms consisting of filaments called hyphae, and they do not contain chlorophyll, and accordingly they cannot carry out the process of metabolism by themselves, so they depend on obtaining their food other living organisms or from the remains of organic materials resulting from the decomposition of dead organisms.

Bacteria

They are microorganisms that differ in shape (rod, spherical, spiral) and most of the bacteria that cause diseases for in plants are rod-shaped and do not produce spores.

Viruses

They are semi-organisms that consist of nucleic acid surrounded by a protein. They are always active inside living cells. They are a group that can only be seen with an electron microscope.

1.12 Classification of plant diseases[1]

We have 2 classes :

- Infectious diseases : characterized by the presence of the pathogen on or in the plant, and sometimes it can be seen with the naked eye or a lens, and is transmitted from one plant to another.

- Non-communicable diseases : in which the pathogen is a non-living factor, i.e. linked to physiological changes such as temperature and humidity... and is not transmitted from one plant to another.

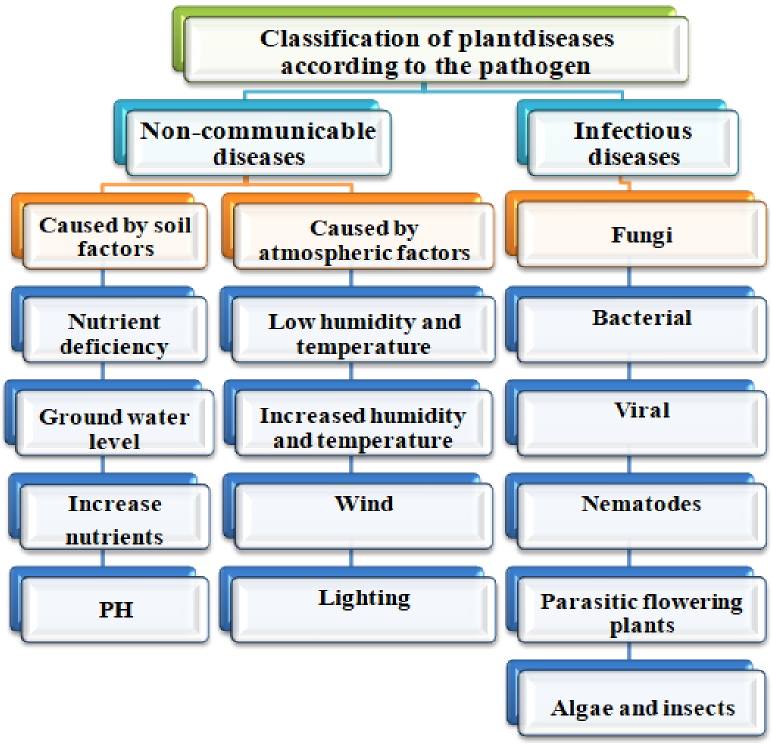


FIG. 1.11 – A chart showing the division of plant diseases according to the pathogen.

1.12.1 Fungal diseases

Seedling fall disease and root rot

The causative fungus spreads in all agricultural areas, as it infects a wide range of crops, especially vegetable crops, in their different stages of growth and different conditions, and from here stems the seriousness of these causes. The disease infects seedlings in seedbeds because the causative fungus is active in conditions of high humidity with poor soil ventilation and lack of lighting. It also infects the field crop plants of tomatoes, peppers, cauliflower, cabbage, squash, cucumbers, melons, beans, peas and strawberries.

Seedling fall disease is caused by fungi of different genera, the most important of which are : Rhizoctonia solani ,Botritis spp ,Phoma spp ,Pythium spp , Fusarium spp ,Alternaria spp .

FIG. 1.12 – Seedling fall disease and root rot. [1]

Early blight

Caused by a fungus Alternaria solani .This disease affects most plants of the Solanaceae family, especially tomatoes, peppers and potatoes.

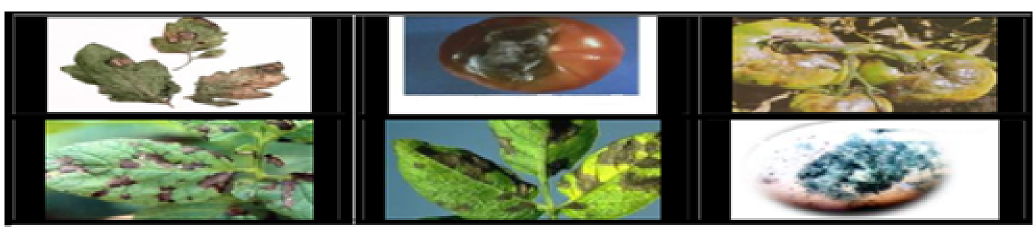


FIG. 1.13 – Early blight (symptoms appear in the form of small dark brown spots that soon turn black, especially on the old lower leaves). [1]

Late blight

It is an ovum caused by Phytophthora infections. The fungus forms a mycelium that grows between the cells inside the tissues and sends its suckers into the cells to obtain its food. Then the spores of the fungus appear through the respiratory stomata and lenticels on the lower surface of the leaves, and they bear lemon-shaped spore sacs. Another bears at its end another sac, and this phenomenon is called proliferation, and for this reason the sporophyte of this fungus is characterized by the presence of sequential bulges that determine the exit points of the sporangia, and the sporangia are dispersed by air.

FIG. 1.14 – Late blight (its color is initially dark brown on the upper surface of the leaf, and there are white fluffy fungal growths on the lower surface, then it turns black, which leads to drying and falling of the leaves, as well as distortion of the fruits) See [1]

Anthracnose

The disease spreads strongly in warm, humid climates , and is caused by the fungus Colletotrichum orbicular and Colletotrichum coccodes. It infects plants of the Solanaceae family. The disease causes wilting of the cotyledonous leaves and scarring at the base of the stem. On adult leaves, pale water-soaked areas appear near the veins that enlarge rapidly and turn gray to brown. The spots may coalesce and form a blight that may lead to the death of the entire leaf.

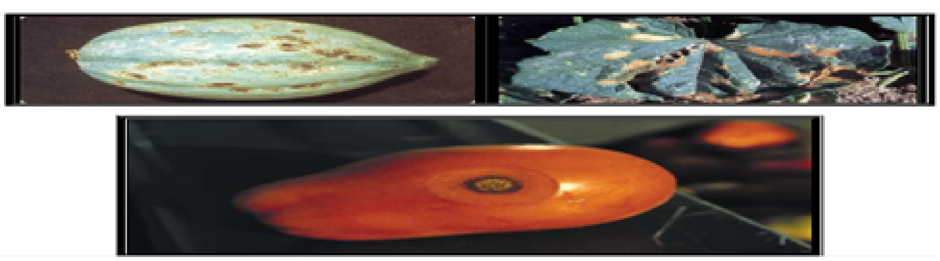


FIG. 1.15 – Anthracnose [1].

1.12.2 Bacterial diseases

Bacterial mold

The cause of the disease is Erwinia carotovora. This disease affects a wide range of vegetables : such as potatoes, carrots, radishes, onions, cucumbers, zucchini, eggplant , tomatoes, cabbage, lettuce, and spinach. This disease is spread all over the world, and it infects vegetables in the field, during transportation, and especially during storage.

Fig. 1.16 – Bacterial mold (the first thing that indicates the presence of the disease is the appearance of soft watery spots, which grow with time, and increase in depth in the affected plant tissue, until all infected parts become soft at the end, and the color of the outer surface of the infected part changes and most parts of the plant that are exposed to this disease They are fruits or storage organs such as tubers. [1].

Bacterial spot

The pathogen is Xanthomaoas victoria. This disease spreads in many regions of the world and mainly affects tomatoes and peppers. The first symptoms that are noticed are the appearance of small dark spots on the leaves and stems of tomatoes and small watery spots. At the beginning of the infection, a white halo appears on these spots, and with the increase in the size of these spots. The halo around it fades, but its center turns brown and becomes slightly recessed from the surface of the fruit.

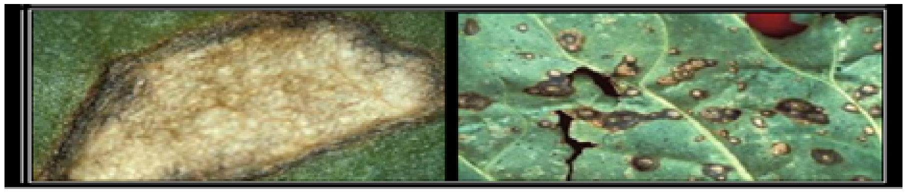


FIG. 1.17 – Bacterial spotting. [1].

1.12.3 Viral diseases

Tomato Mosaic

Pathogen is the Tomato Mosaic virus. This disease is spread all over the world, and infection with it reduces the production of tomatoes grown in the field or glass or plastic houses. This disease also affects pepper. It means irregularity of Green discoloration, i.e. the appearance of pale green spots alternating with dark green spaces, without clear borders, but the affected plants are stunted, and their leaves develop abnormally, and sometimes the leaves are shortened and take the form of a thin thread. Symptoms on the fruits during the late period of their maturity in the form of brown contamination inside the fruit, these symptoms appear on the stem in the form of lines along the stem, and it is called the stripping of the stem.

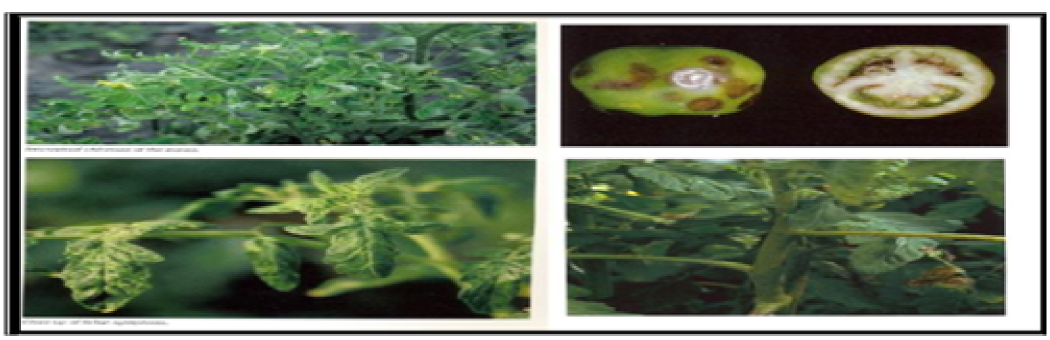


FIG. 1.18 – Tomato mosaic. [1].

Watermelon Mosaic

It is caused by the Watermelon Mosaic Virus, which is one of the viruses transmitted by the aphid insect that leads to the occurrence of clear mottled leaves, dwarfing of the plant, the small size of the young leaves and their deformation. The infection also leads to the emergence of small surface growths raised above the leaves.

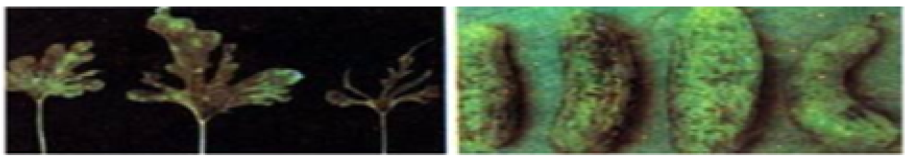


FIG. 1.19 – Watermelon mosaic. [1].

1.13 Agricultural disturbances[1].

It is any deviation in the behavior of the plant resulting in the appearance of disease symptoms under the influence of changes in environmental conditions, which negatively affects the nature of the growth and development of the plant and thus on the crop output.

It is called a physiological disease and is characterized by the fact that it is not a collective infection, meaning that all plants in the field are infected with it.

1.13.1 The effect of environmental factors on the plant in disease events

- The environmental conditions are not suitable for the growth and activity of the parasite and suitable for the growth and development of the plant - in this case, no disease occurs.
- The environmental conditions are not suitable for the growth and activity of the parasite, and also unsuitable for the growth and development of the plant in this case a non-infectious disease occurs physiological damage.

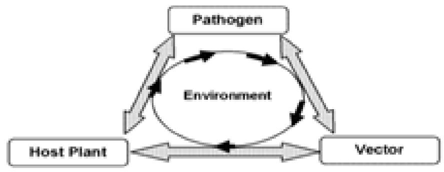


FIG. 1.20 – Image illustrating the relationship between the pathogen, host and vector in the presence of environmental conditions [2].

- The environmental conditions are suitable for the growth and activity of the parasite, and also suitable for the growth and development of the plant in this case infectious disease viral, insect, rodent, parasitic flower plants, across, and snake worms.
- The environmental conditions are suitable for the growth and activity of the parasite, and are not suitable for the growth and development of the plant in this case, a complex disease or several overlapping diseases occurs.

1.13.2 The most important environmental factors affecting plant growth and development

1 Weather factors :

Temperature

It is a primary driver of insect development, affecting their metabolic rate and population growth [11]. Thus, the duration of an insect's life cycle is highly influenced by the number of days when the temperature is suitable for its development. Two temperature thresholds can be defined : an upper threshold, in which insect development slows down, or stops, and a lower one where there is no insect growth. These thresholds vary according to the specific insect species [12]. A degree day is a concept concerning the accumulation of heat by insects [13]. One degree day is a period of 24 h in which the temperature was one degree above a given baseline [12].

Atmospheric humidity

It is a favorable condition for the development of fungus diseases. It can be caused by the weather or by poor watering practices that cause a high wetness among the leaves e.g., leaf mold or bacterial spot [16].

Lighting

Plants absorb part of the radiation coming from the sun and reflect the rest. Depending on the health of the plant, the amount of radiation absorbed and reflected differs. This difference can be used to distinguish between healthy and diseased plants and to assess the severity of the damage [18]. Diseased plants with damaged leaves have different leaf spectral reflectance compared to healthy plants because of the different chlorophyll concentrations and leaf tissue damage. Diseased plants end up absorbing less of the visible light and more of the NIR (Near Infra-Red) light. From this knowledge, disease detection can be done using leaf reflectance information [18, 14, 15].

RAINFALL[17]

The maximum monthly rainfall is the maximum rainfall on any day for a particular month and is given by :

$$R\max = MaxRxi.......................................(1)$$

Where "Rmax" is the maximum monthly rainfall, "Rxi" is the maximum daily rainfall.

Disease D	Rainfall	Ambient Temperature	Ambient Humidity	Soil Moisture
Powdery mildew	<1%	10-20°C	90-100%	10-14%
Anthracnose	<1%	10-15°C	80-100%	10-14%
Rust	0.5-1%	21-26°C	75-100%	10-14%
Root rot leaf blight	>0.5%	>30°C	>80%	10-14%

FIG. 1.21 – This table presents some numerical Range of features for a particular disease.

2 Soil factors : Soil PH, Nutrients.

3 Environmental pollution with chemical pollutants and pesticides.

1.13.3 The most important physiological diseases affecting vegetable crops

Cracks

Where cracks appear on the fruits, whether superficial or recessed, they may be longitudinal, circular or transverse. One of the causes of this phenomenon is the occurrence of an imbalance in the water balance inside the plant for any reason, such as irregular irrigation, as well as an increase in nitrogen fertilization and a decrease in potassium fertilization leads to the occurrence of this phenomenon, and it also occurs when using sensitive varieties with large fruits, thin-skinned.

FIG. 1.22 – Circulacrack , transverse crack and recessed crack [1].

Sun blight

The fruits are infected with it when they are directly exposed to strong sunburn, as this leads to an increase in the temperature of the tissue facing the sun and it turns white or yellow, and it continues in this situation while the rest of the fruit is naturally colored.

FIG. 1.23 – Sun blight. [1].

The central gaps in the seeds

Brown gaps appear in the center of the seeds in the cotyledons, and they can be seen when separating the cotyledons from each other, and the reason for this phenomenon is the deficiency of the manganese element, and it abounds in the case of alkaline soils, where there is a deficiency of the manganese element that facilitates plants.

FIG. 1.24 – The central gaps in the seeds . [1].

Irregular discoloration of fruits or spots

Where spots of a color contrary to the natural color of the fruit appear green, yellow or red, and three types of tissues may appear from the inside : a tissue of natural color, white fabric, and brown fabric. Potassium deficiency is the main cause of this phenomenon. It also appears in the case of boron deficiency.

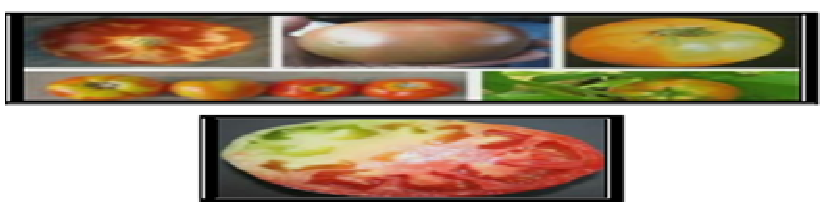


FIG. 1.25 – Irregular fruit discoloration or spots. [1].

Deficiency of nutrients

Due to the intensive and stressful nature of soil cultivation within greenhouses, symptoms of deficiency in elements that are consumed quickly and that require constant replacement are often observed in plants growing in them. In the soil, negative effects are sometimes equivalent to the effects of its deficiency [1]

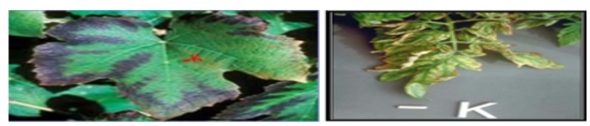
Nutrients Necessary for Plants

- Basic Elements like Carbon – Hydrogen – Oxygen.
- Major Necessary Elements like Nitrogen – Phosphorus – Potassium.
- Medium Necessary Elements like Calcium – Sulfur – Magnesium.
- Minor Necessary Elements like Iron – Zinc – Manganese – Copper – Boron - Molybdenum – Chlorine.

Name	Chemical symbol	Relative % in plant to N	Function in plant
Primary macronutrients			
Nitrogen	N	100	Proteins, amino acids
Phosphorus	P	6	Nucleic acids, ATP
Potassium	K	25	Catalyst, ion transport
Secondary macronutrients			
Calcium	Ca	12.5	Cell wall component
Magnesium	Mg	8	Part of chlorophyll
Sulfur	S	3	Amino acids
Iron	Fe	0.2	Chlorophyll synthesis
Micronutrients			
Copper	Cu	0.01	Component of enzymes
Manganese	Mn	0.1	Oxygen evolution
Zinc	Zn	0.03	Activates enzymes
Boron	B	0.2	Cell wall component
Chlorine	Cl	0.3	Photosynthesis reactions
Molybdenum	Mo	0.0001	Nitrogen fixation

FIG. 1.26 – Nutrients and their role.

This table represents some nutrients and their role :

Element	Role in the plant	Symptoms
phosphorus	Necessary for the formation of seeds and is of great importance in the growth of roots and the ripening of seeds and fruits	•Plant growth weakens and the leaves are bluish-green,and the lower leaves turn into a light ,bronze color with purple or brown spots.The twigs are short,slender,and spindle-shaped.  Phosphorus deficiency. [1]
potassium	Works as a catalyst in many reactions.•Responsible for opening and closing stomata.Works on the transfer of food from leaves to fruits.Responsible for the osmotic regulation of the cell.	•The plants haves lender branches,and in severe cases,the phenomenon of regressive death(death of the tops)appears on them.  Potassium deficiency. [1, 19]
Nitrogen	It is found in most of the components of cells where protein is found.	The stems are short and cylindrical.  Symptoms of nitrogen deficiency. [1]

Tab. 1.1 – Some nutrients and their role

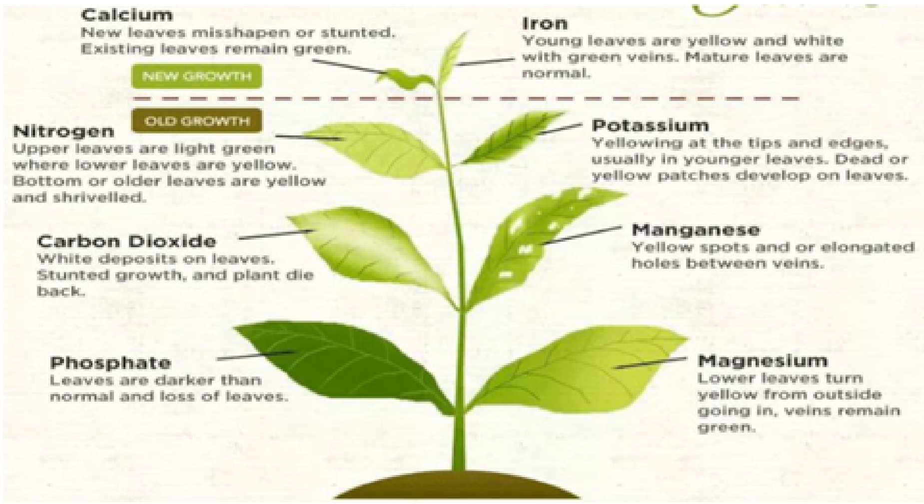


FIG. 1.27 – Places of appearance of symptoms of nutrient deficiency on the plant . [20]

1.14 The disease severity

When evaluating the severity of a plant disease, a scale known as a disease interval (or category) scale is used. For plant diseases, this scale is often based on the percentage of the affected area that exhibits symptoms. Quantitative interval scales are frequently used in studies of plant disease and plant breeding. A rater's assessment of the disease severity is used to calculate the disease severity index (DSI), which is expressed as a percentage and is calculated as follows :

DSI (%) = [sum (class frequency score of rating class)] / [(total number of plants) (maximal disease index)] 100. ....(2)

However, relatively few research have looked into how different scales affect the DSI's accuracy.[22] These factors affect how severe a plant disease is :

• The kind of plant (some hybrids are more delicate than others) ;

• The number of infectious spores in the inoculums (crop rotation)

• Agricultural management (overwintering on crop wastes, tillage) ; environmental circumstances (e.g., severely rain distributes spores across a wider region, extended-lasting leaf wetness period) ;

The weather conditions are the primary determinant of the fungal pathogen's ability to infect. Rain, high relative humidity, wet leaves, mist, or fog all encourage illness, and warm temperatures during these wet times hasten the infection process.[21]

1.15 Disease monitoring[21]

The danger of an infection in the field is determined hourly by weather stations with sensors for rain, relative humidity, leaf wetness, solar radiation, soil parameters, and temperature. Applications of fungicides may be timed best using this knowledge. For instance, the history of the field (the number of spores present) and the climatic conditions during fluorescence have an impact on the Fusarium Head Blight epidemic. If the weather is unfavorable during bloom, it won't happen. However, if the right conditions exist for an infection to take hold during bloom, the flower will become infected. This illness results in "deaf ears," malformed grains, and an increase in mycotoxins in the crop.

1.16 Percentage Scales[23]

• This often keeps track of the number of plants or organs that fall into recognized percentage illness groupings.

• Based on the percentage of damage visible to the human eye, illness groups are categorized.

• Horsfall and Baratt (1945) proposed a 12-grade scale after accounting for the fact that the grades recognized by the human eye are roughly equal divisions on a logarithmic scale and typically follow the Weber-Fechner law, which states that visual activity depends on the logarithm of the stimulus intensity.

• In terms of percentage disease evaluation, the eye evaluates the healthy area above 50% and the sick area up to 50%.

The categories of the Horsfall and Baratt grading system were as follows :

1= 0%, 2= 0-3%, 3= 3-6%, 4= 6-12%, 5= 12-25%, 6= 25-50%, 7= 50-75%, 8 = 75-87%, 9 = 87-94 %, 10= 94-97%, 11= 97-100 %, 12 =100% disease.

•Because pathogens reproduce at a logarithmic rate, this logarithmic scale is suitable for loss estimation as well as illness assessment and epidemiological investigations.

•The British Mycological Society created a technique utilizing a percentage scale (Anon, 1947) to gauge late blight on potatoes.

•The average infection or infection index, often also known as disease severity index (DSI) or percent disease severity index (DSI(%)), is typically computed using the following formula :

$$DSI(\%) = \frac{\sum(class\_frequency*score\_of\_rating\_class)}{(Total\_number\_of\_observations)*(\max imal\_disease\_index)}*100 \ldots\ldots(3)[24]$$

$$DSI = \frac{\sum(class\_frequency*score\_of\_rating\_class)}{(Total\_number\_of\_observations)} \ldots\ldots(4)[24]$$

• Severity estimates from fairly small areas can be combined to cover large areas, or can be defined as the area or volume of plant tissue that is diseased [18], viz., village, district or state. This overall index can be obtained by using the formula :

$$Percent\_Disease\_Index(PDI) = \frac{Field\_rating\_class*Number\_of\_hectares\_in\_the\_class}{Total\_number\_of\_hectares} \ldots\ldots(5)$$

The percentage scales have many advantages such as :

- The scale is flexible in that it can be divided and subdivided conveniently.
- It is universally known and can be used to record both the number of plants infected (incidence) and the area damaged (severity) by a foliage or root pathogen.

1.17 Computer Simulation[25]

One of the recent developments is the use of systems analysis to produce simulation models of some plant diseases. Systems analysis attempts to aggregate all the factors influencing the development of specific plant diseases into a computer-based model, and also attempts to model the complexity of the many interactions among these factors to accurately predict plant disease.

Simulation systems depend on basic (biological) information that must be obtained before developing the model. About fungal diseases, the necessary information includes : the effects of rainfall , temperature , humidity , wind , lighting , solar radiation and clouds on the development of sporangia or conidia, sporulation development , duration of spore production , number of spores produced , sporulation dispersal , sporulation germination and injury occurred. This tool is used in combination with related differential equations to determine key dynamic features of the pathosystem.

Several systems have been used to simulate plant disease dynamics to prevent or control disease occurrence like EPIDEM program , which was prepared by Wagonner and Berger , is one of the first programs . It was designed to simulate the epidemiology of early blight on tomatoes and potatoes caused by the fungus Alternaria solani.

1.18 Conclusion

In conclusion, plant diseases pose a serious problem for the agricultural industry, affecting sustainability and global food production. Economic losses decreased agricultural yields, and food poverty may result from these diseases. However, there is a chance to lessen the effects of plant diseases via research, technical improvements, and efficient disease control techniques.

An all-encompassing strategy that combines precautions, early detection, precise diagnosis, and effective management methods is needed to combat plant diseases. Reduced disease incidence and pathogen dissemination are achieved via the use of disease-resistant crop types, integrated pest control techniques, crop rotation, sanitation, and these practices.

Agriculture's approach to managing illness is evolving because to cutting-edge technology like artificial intelligence (AI), the internet of things (IoT), and remote sensing. Large data sets may be analyzed by AI algorithms, which can then be used to diagnose diseases, estimate disease risks, and forecast epidemics. Real-time monitoring, data collecting, and targeted treatments are made possible by IoT devices and sensors, improving the effectiveness and precision of disease control.

To effectively manage plant diseases, cooperation between researchers, farmers, legislators, and agricultural professionals is essential. To reduce the impact of plant diseases on agricultural systems, it is crucial to share information, put best practices into practice, and promote sustainable and ecologically friendly methods.

Despite persisting difficulties, technological, scientific, and disease control advances give reason for optimism regarding sustainable agriculture, better crop resilience, and enhanced food security. We may work toward a future in which plant diseases are successfully managed and the world's food production is protected by constantly advancing our knowledge of plant diseases and implementing cutting-edge remedies.

Chapter 2

PHASE-2- AI and IoT in Plant Diseases

2.1 Introduction

Plant diseases, which also result in crop losses, decreased yields, and financial hardship for farmers, drastically reduce agricultural productivity. The Internet of Things (IoT) and artificial intelligence (AI) have the potential to dramatically change how diseases are managed in the agriculture industry.

Artificial intelligence (AI) has made it possible for computers to do activities that previously required human intelligence, such as pattern recognition, decision-making, and problem-solving. The Internet of Things, on the other hand, entails connecting actual objects and sensors so they can communicate and gather data.

AI and IoT have various benefits when used to address plant diseases. For instance, smart sensors buried in soil may track temperature, humidity, soil moisture, and air quality, among other environmental variables. By spotting trends and departures from the norm, this data along with AI algorithms might enable the early diagnosis of disease epidemics.

To give precise disease risk estimates, AI-powered systems can examine vast amounts of data, including satellite images, weather patterns, and historical illness records. This lessens the need for broad-spectrum insecticides and has a less negative impact on the environment by assisting farmers in making educated decisions about disease prevention, crop protection strategies, and focused treatments.

AI may also assist in the identification of diseases by examining photos of samples of sick plants and comparing them to huge databases of disease signs. This may enable rapid and exact pathogen detection, enabling accurate and prompt treatment.

AI and IoT device integration can allow for real-time monitoring and administration of illness management procedures. Soil moisture levels, for instance, can be used to alter automatic irrigation systems, maximizing water use and lowering the risk of waterborne infections. Similar to autonomous cars, AI-powered drones with sensors and AI algorithms can scan vast fields, locate illness hotspots, and focus on certain regions for intervention.

In addition, AI-powered prediction models can identify disease outbreaks and provide farmers advance notice, enabling proactive disease control techniques. This makes it possible for farmers to reduce disease risks by implementing preventative measures including modifying planting schedules, improving crop rotation, and using disease-resistant cultivars.

It is crucial to remember that effective data collecting, data exchange, and dependable connection are necessary for the successful application of AI and IoT in the control of plant diseases. For these technologies to be widely adopted and used effectively, it is essential to address data privacy issues, ensure device and system compatibility, and offer training and assistance to farmers.

IoT and AI together have enormous potential to revolutionize plant disease control in agriculture, to sum up. Farmers may improve disease diagnosis, prevention, and control via the use of real-time data, predictive analytics, and autonomous systems, which will result in healthier crops, higher yields, and more environmentally friendly agricultural methods. An important step toward developing effective and resilient agricultural systems against plant diseases is the integration of these technologies.

In this section we represent some definitions of IA and IoT.

Problem Statement

In this study, we use the data and trends related to agriculture for the prediction of plant diseases. The objective is to create a model/ classifier that can predict diseases by taking different parameters like temperature, humidity, leaf image etc... However the suggested DL classifier gives output in optimized time with good accuracy.

2.2 The objective of this work

- We carry out this effort, which is regarded as a link between artificial intelligence and agriculture, to address the issue of plant diseases or the so-called nightmare of farmers and to progress the national economy.

- Utilize online monitoring tools to save time and money.
- Give your crops the chance to grow in the ideal circumstances.
- If automated detection is employed, it will be quicker, more accurate, and need fewer resources.
- A weather prediction to help you decide what to do and when.
- Resources for planning your work to increase productivity.
- Improving field-wide homogeneity and production efficiency.
- Improving agricultural quality, lowering environmental impact, and reducing risk from an economic and environmental standpoint.
- Reduces the number of plant fatalities by :

- Prediction of the plant illness aids in reducing plant fatalities.
- keeping an eye on the surroundings of the plant.
- Giving the user feedback on potential actions to take to rescue the plant.

2.3 The related work

This section deals with a detailed analysis of existing methods with their drawbacks. This systematic literature review helps us to understand the application of artificial intelligence approaches in plant disease prediction :

- In [27] P.Wang et al., (2021) developed for plant disease classification along with the sensor data. In this model, a camera model-equipped controller is designed with moisture, color, texture, humidity, and temperature of the leaf. However, this method suffers from the high computational complexity.
- Further, a hybrid convolutional neural network (HCNN) [28] is trained with a dual image database by J.Arshad et al., (2022). The database consists of previously infected images, which are used for training the database for such diseases. Secondly, texture, color, and morphology features are extracted from the image. However, this method consumed

higher training time for feature training.

- In [29], M.Cicioglu et al., (2021) integrated DL with IoT for automatic disease identification from plants. The IoT is used for remote sensing of field parameters storage, with a modified ResNET51 model which was used on the cloud to build smart disease detection. This method suffers from low classification performance.
- Further, DeepLens [30] variations are introduced for continuous monitoring of data with ubiquitous access and reliability, which is accessed by cloud data integrating with recursive CNN classification by H.N.Saha et al., (2021). The RCNN is used to identify the condition of leaves of fruit trees and vegetable plan. However, this method is not useful for diagnosis of hand full of plants and trees diseases detection.
- In addition, AI and IoT-enabled smart agriculture technologies' [31] system is developed with decision tree classification by P.S. Chatterjee et al., (2021). The data from the hardware is processed by AI, which contains valuable data for the prediction of all the parameters of crop. However, this method suffers from power-related issues in a real-time environment.
- In [32], Ramakrishnan Raju et al., (2022) implemented An AI-SHES developed with a user-friendly environment for farmers using a Raspberry Pi controller, IoT environment with AgriHydroponic application. The farmers monitor and control their hydroponics farm field using the application with manual and automatic controlling modes of operation. An artificial intelligence system is placed across the cloud served with DLCNN, which continuously monitors the sensor data and plant disease status and sends the necessary alerts to the farmers using.
- Gayathri et al., (2021) [33] used the Internet of Things (IoT) and Machine learning algorithms such as SVM and CNN to monitor and detect crop disease. This model performs a comparative analysis of SVM, CNN, naive Bayes, and KNN.
- In [34], K.Abhirami et al., (2021) develop a methodology that combines IoT and Image processing and performs classification using a DL model that helps in crop disease

prediction and thereby supports increased productivity.

- In [35], Reddy et al., (2023) used two models, microcontrollers and sensors, to acquire real-time data, the work reported in this paper concerns the identification of plant diseases. The first model is a machine learning model that was taught using the Random Forest method, while the second model is a deep learning model that was taught using a (CNN). The accuracy of both the CNN model and the random forest model is 95.99%. The accuracy of the CNN model is 99.2%.
- In [36] S. Ramesh et al., (2018) present a technique to spot the diseases in initial phases and alert the farmers.
- In [37], Ferentinos et al. (2018) used a publicly accessible collection of 87,848 photos that included 25 different plant species in 58 different classes of [plant, illness] pairs, including healthy plants. The best performance of the trained CNN models, which included several different architectures, was 99.53%.
- Sladojevic et al. (2016) used Deep CNN in [38] to detect plant illnesses using photographs of leaves utilizing a comparable quantity of online data, which comprised fewer diseases (13) and more plants (5). Their models' success percentages ranged from 91% to 98%, depending on the testing results.
- In [39], Mohanty et al. (2016) examined the diagnosis of 26 plant diseases using two well-known and established CNN architectures (AlexNet, GoogLeNet).utilizing pictures of the leaves from 14 different plants from a public database. Their findings were highly encouraging, with automatic recognition success rates reaching 99.35%.

2.4 Artificial intelligence (AI)[40]

It was defined by McCarthy, 1988 as :

"The goal of artificial intelligence (A.I.) is machines more capable than humans at solving problems and achieving goals requiring intelligence. There has been some useful success, but the ultimate goal still requires major conceptual advances and is probably far off. There are three ways of attacking the goal. The first is to imitate the human nervous system. The second is to study the psychology of human intelligence. The third is to understand the common sense world in which people achieve their goals and develop intelligent computer programs. This last one is the computer science approach."

2.5 Machine learning (ML)[41]

Programming computers to learn from data is the practice. Data used as examples or training sets in machine learning.

2.5.1 Types of Systems of Machine Learning

There are different types of ML systems. We can divide them into :

Supervised Learning

In this type the data that you feed into the algorithm, with the desired solution, are referred to as "labels."

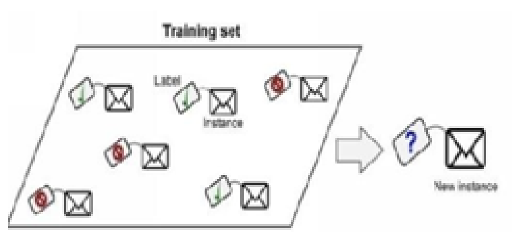


Fig. 2.1 – Supervised learning example. [41]

The most important supervised algorithms

- K-nears neighbors.

- Linear regression.

- Neural networks.

- Support vector machines.

- Logistic regression.

- Decision trees and random forests.

Unsupervised Learning

In this type you can guess that the data is unlabeled.

FIG. 2.2 – Unsupervised learning example . [41]

The most important unsupervised algorithms

- Clustering : k-means, hierarchical cluster analysis.

- Association rule learning : Eclat, apriori.

- Visualization and dimensionality reduction : kernel PCA, t-distributed,PCA.

Semi-Supervised Learning

In this type you can guess that the data is labeled/unlabeled.

Reinforcement Learning

It is another type of ML system. An agent "AI system" will observe the environment, perform given actions, and then receive t rewards in return. With this type, the agent must learn by itself. Ties called a policy. You can find this type of learning type in many robotics applications that learn how to walk

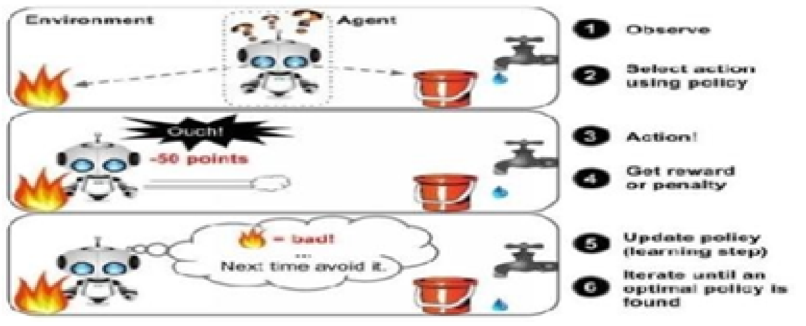


FIG. 2.3 – Reinforcement learning example [41]

2.6 Machine Learning Models (ML)[12]

After data is processed and features are extracted, models can be used for classification, and regression, among other goals.

In classification, a new data sample is assigned a label according to the relations retrieved during the training process.

In regression, a continuous output value is estimated from the input variables.

These some ML models :

2.6.1 Support Vector Machine (SVM)[26]

One of the most popular supervised learning algorithms is SVM, which may be used for both classification and regression problems. The goal of the SVM is to find the best decision boundary or line to split n-dimensional space into classes so that the following data points may be swiftly classified. The name of this best-choice boundary is a hyperplane. The extreme vectors and points that SVM chooses to use in the development of hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as SVM.

The above-mentioned method is applied when our data is linearly separable. If the data is not linearly separable, then we use various kernel functions to modify our data into higher

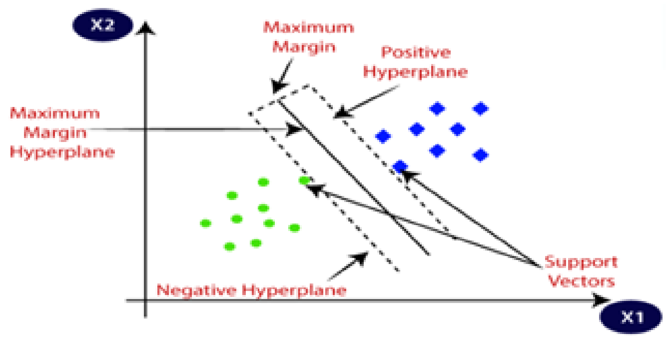


FIG. 2.4 – Support Vector Machine. [42]

feature

space so that they can be linearly separable. Some of the kernel functions used are :

- **Linear kernel:** $K(x_i, x_j) = x_i^T x_j$
- **Polynomial kernel:** $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
- **RBF kernel :** $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0$
- **Sigmoid kernel:** $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

Here, γ, r and d are kernel parameters.

2.6.2 Naive Bayes

It is a classification technique that relies on the Bayes theorem and makes the assumption that characteristics are independent of one another. The reason it is referred to be "naive" is because it assumes that the presence or absence of one characteristic in a class has nothing to do with the presence or absence of other features. Naive Bayes has been widely utilized and shown to be effective in many applications, particularly in text categorization and spam filtering, despite its oversimplifying premise. By multiplying the individual probabilities of each feature existing in that class, it determines the likelihood that a given instance belongs to a certain class. Naive Bayes is renowned for being straightforward, quick, and effective at handling enormous feature spaces.

2.6.3 K-Nearest Neighbor (KNN)

It is the easiest-implemented ML algorithm under the supervised learning category. KNN assumes that similar objects are close together. Prediction and classification of the target value are performed based on stored data and distance functions like Makowski, Manhattan, Euclidean, etc. Distance between input or the sample to be predicted and training points are evaluated. Points having the smallest distance (as the name suggests, k- nearest neighbors) are considered. The target value is thus obtained by adding all these selected k neighbors. The only drawback of KNN is having high time and space complexity. This is mainly due to the use of all dataset samples every time while predicting [118].

2.7 Deep learning (DL)

DL is a class of ML that exploit many layers of non-linear information processing for supervised or unsupervised feature extraction and transformation, and for pattern analysis and classification [43]. And the concept of DL originated from a paper published in Science by Hinton et al. [44] in 2006. The basic idea of DL is : to use ANN that contains a quite large number of processing layers [45], for data analysis and feature learning, data features are extracted by multiple hidden layers, and each hidden layer can be regarded as a perceptron, the perceptron is used to extract low-level features, and then combine low-level features to obtain abstract high-level features, which can significantly alleviate the problem of local minimum. It has now been successfully applied in computer vision, pattern recognition, speech recognition, natural language processing and recommendation systems [46] and depends on experience and luck, and cannot automatically learn and extract features from the original image [47].

Additionally, it has been widely applied in many sectors of the world such as Business, agriculture, the automotive industry etc [43]. At present, DL methods have developed many well-known deep neural network models, including deep belief network (DBN), deep

Boltzmann machine (DBM), stack de-noising autoencoder (SDAE) and deep CNN [48]. The performance of DL approaches has been improved by undertaking various techniques such as stochastic gradient descent (SGD), batch normalization, and dropout.

2.7.1 Batch Normalization[43]

It helps to minimize the challenges posed by internal Covariate Shift. The input of each layer is normalized by adjusting the mean and variance of the input across one mini-batch. It allows the use of much higher learning rates and less worry about initialization, and in some cases eliminates the need for Dropout. It potentially helps in two ways : faster learning and higher Accuracy is overall [98]. In each experiment, batch normalization and the ReLU activation function are used.

We now explore the evolution of deep learning :

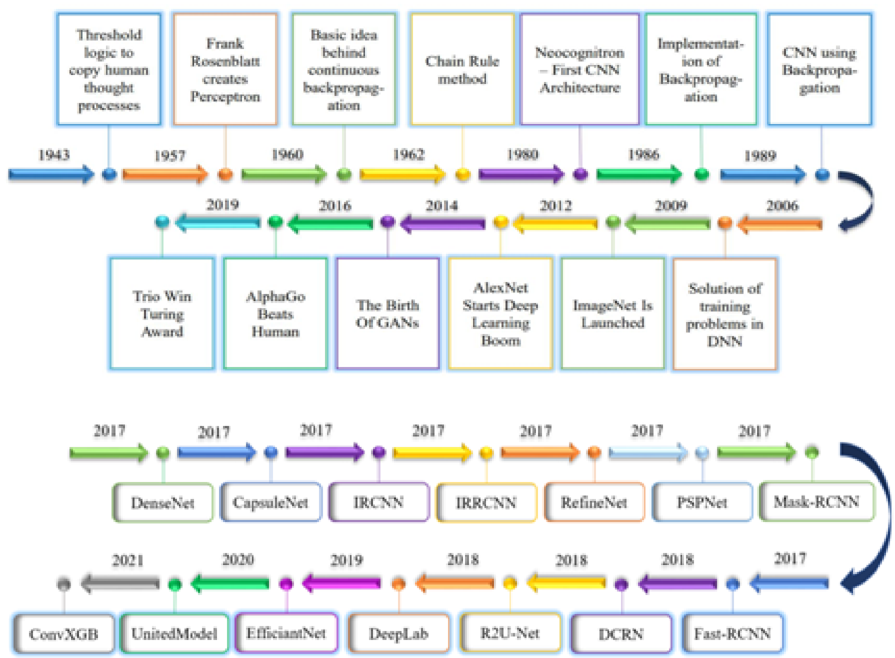


FIG. 2.5 – Summary of the deep learning evolution from 1943 to 2021 See [49]

2.8 Deep Learning models (DL)

Some of the DL approaches are given below :

2.8.1 Artificial Neural Networks (ANN)[12]

ANN are models inspired by biological brains. ANN consists of neurons distributed in input, one or multiple hidden, and output layers and can have multiple units in each layer. With more hidden layers, an ANN can learn complex relations from the hierarchical combination of multiple features, and thus create high-order features. DL is associated with ANNs that contain a large number of layers. Moreover, to initiate the process, initial weights are assigned randomly. Learning occurs by a process called optimization, which is an iterative method for minimizing an error function, typically based on the Gradient Descent algorithm. Instead of calculating the gradient from the entire dataset, the optimization process typically uses chunks of data records called batches. After the network processes the input, the output is compared to the expected output and the error is computed. The error is then propagated back through the network, one layer at a time, and the weights are updated according to the amount they contributed to the error. This updating process is called back-propagation. After all records in the dataset are processed once, a training epoch is completed training the network can require several epochs until desired results are achieved.

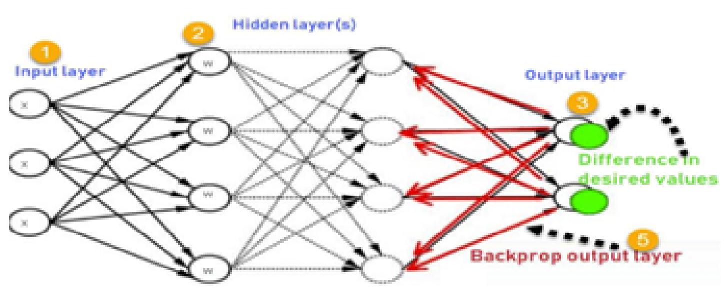


FIG. 2.6 – ANN example [50]

2.8.2 Convolutional Neural Networks (CNN)

CNNs are a type of ANN, typically one of the most powerful techniques for modeling complex processes [45] used in DL for image recognition, text recognition, object recognition, and classification. CNNs are used in DL to recognize objects within an image or text. It is a commonly employed technology in machine vision applications. Additionally known as "convnents" and "CNN," The subset of deep neural networks responsible for doing visual data analysis. When attempting to identify items based on image and video data, this design style is scrutinized. It is used in applications such as video or image recognition, neural language processing (NLP), and other types of processing [51]. The CNN algorithm includes four distinct layers. Their respective names are the Convolution Layer, ReLU Layer, Pooling Layer, and Fully Connected Layer.[35]

Convolution Layer[35]

It makes it possible to extract characteristics hierarchically. It features a large number of filters that can execute convolutional operations. The mathematical process known as "convolution" produces a third function that illustrates how the forms of two functions interact to produce a third function. Similar to a convolution procedure, CNN emphasizes measurements that are near together while taking many measurements. As a result, we are currently conducting updated measures, the results of which are a weighted average of the prior measurements, with a preference for the more recent data over the older ones. A feature map will be created from the filter input.

Relu Layer[35]

After the feature maps have been acquired, the final step is to move them to a ReLU layer. The rectified linear unit, often known as the "ReLU," is not a discrete component of the CNN process. The convolution procedure, which was covered in the last tutorial, must be finished with an additional step. Some instructors and writers discuss both phases

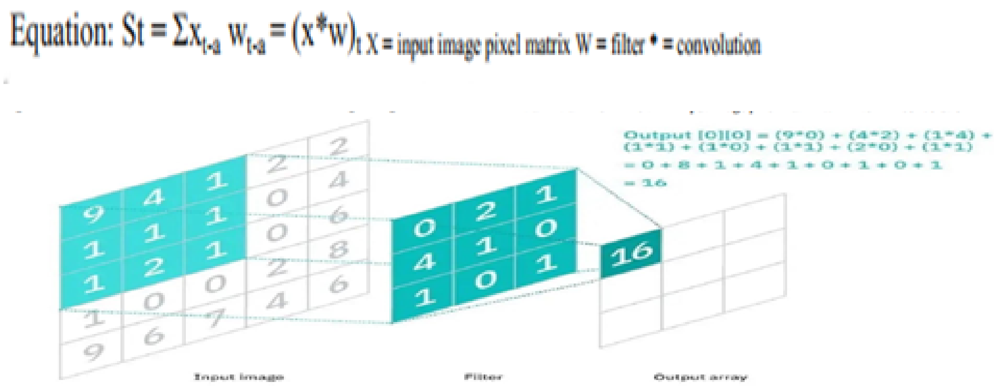


FIG. 2.7 – Convolution Layer [35]

individually. The application of the rectifier function will raise the level of nonlinearity in our photographs. The fact that pictures are intrinsically non-linear is the driving force behind this objective. When you examine an image closely, you will notice that it contains numerous elements that are not organized linearly. To compensate for any linearity that may be injected into an image as a result of the convolution technique, the rectifier serves to further distort the image's existing linearity. Observing the following image, we can observe its transformation as it undergoes the convolution operation, followed by the rectification operation. This will assist us in comprehending how the process operates.

FIG. 2.8 – Relu Layer [35]

Pooling Layer[35]

It is now added after the rectified feature map has been processed. A procedure called "pooling," which is a sort of downsampling, can be used to lower the dimension of the feature map. The pooling layer is another one of CNN's structural components. Dimensionality reduction is the principal purpose it serves. This is one of the most effective methods for mitigating the issue of overfitting. Using a variety of filters, it may also be used to identify the edges and corners of objects. There are two distinct methods of resource pooling.

• **Max pooling**it takes the value with the largest filter size.

• **Average Pooling**The average value for each block of values will be determined by using the avg pooling method.

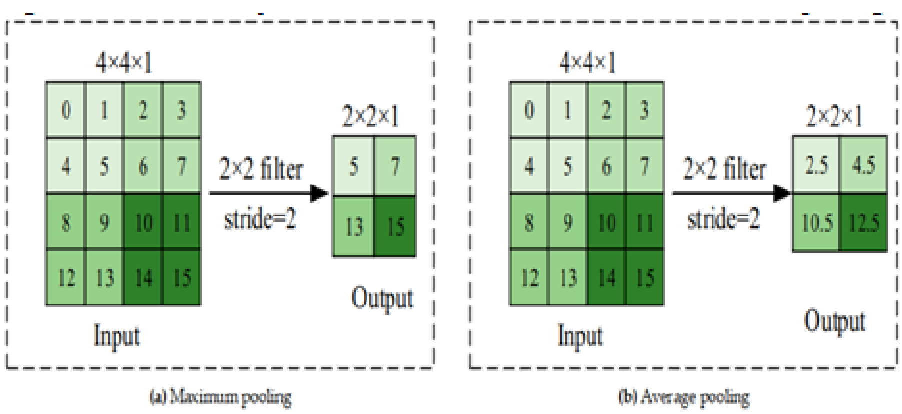


FIG. 2.9 – The process of pooling operation. [52]

Output Layer[35]

This consists of a layer that is fully connected, which is then followed by an activation function called SoftMax, which determines the output classes.

Fully Connected Layer[35]

In the same way that the output of a neural network would be sent to fully connected layers, the output matrix of this layer is transformed into vector form and then sent to those layers. While neurons in the layer below are responsible for weights and bias, this layer assigns vectors to each neuron. It generates the output matrix and the last layer to produce the output image and its class name. Using a two-layer design, the CNN output is provided by both the SoftMax and Logistic layers. The logistic layer handles both binary and multiclass classifications, whereas the SoftMax layer handles multiclass classifications. This is CNN's general architecture and transfer learning is utilized in the retraining of CNN's networks.

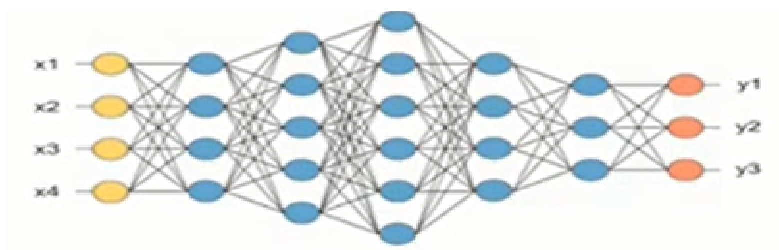


FIG. 2.10 – Fully connected layer. [35]

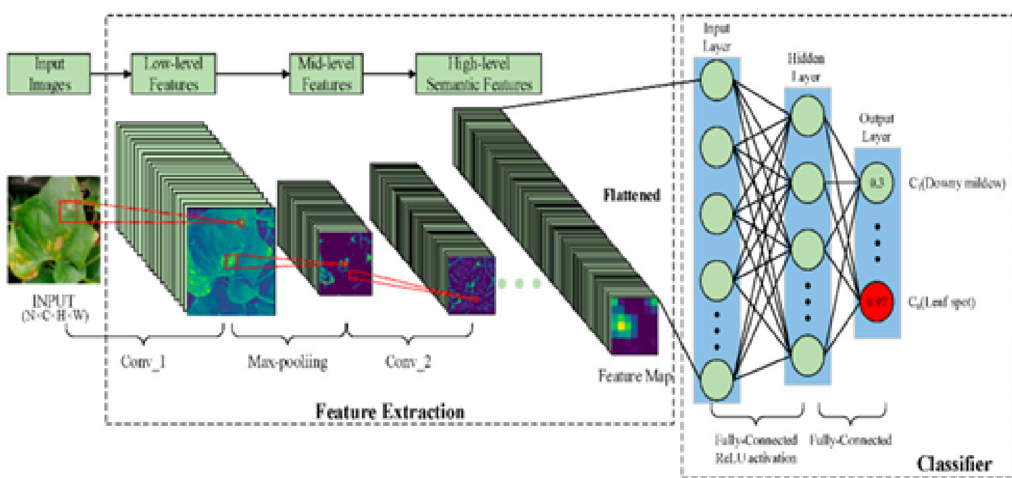


FIG. 2.11 – A typical CNN architecture for plant disease classification [52]

2.8.3 Convolutional Neural Network Architectures[12]

Image classification has achieved great results, with various model architectures being developed over the last 10 years. Most of these DL models were proposed in the context of the "Large Scale Visual Recognition Challenge" (ILSVRC). These models include well-known architectures such as AlexNet, GoogleNet, VGG, and ResNet, AlexNetOWTBn, Overfeat, which have been widely used for image classification in different application domains.

ResNet[43]

He et al. in their paper He et al., (2015) [102] introduced the ResNet model which was a basis of the ILSVRC 2015 and COCO 2015 classification challenge. Their model won 1st place with an error rate of 3.57% in the ImageNet classification. The inability of multiple non-linear layers to learn identity mappings and degradation problems motivated the deep residual learning framework (ResNet).

ResNet is a network-in-network (NIN) architecture that relies on many stacked residual units. These residual units are the set of building blocks used to construct the network. A collection of residual units forms building blocks that leads to the ResNet Architecture [102]. The residual units are composed of convolution, pooling, and layers.

A further update of ResNet was done to obtain more accuracy by updating the residual module to use identity mappings. A ResNet model with 50,101 and 152 layers and load it with pre-trained weights from ImageNet was created. Finally, a customized softmax layer was created for the task of plant disease identification. There are several types of ResNet, including ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-110, ResNet-152, ResNet-164, and ResNet-1202.

MobileNetV2

It is a CNN architecture designed for efficient and lightweight deep learning on mobile and embedded devices. It is an improved version of MobileNetV1, incorporating several key enhancements to achieve higher accuracy and better performance while maintaining low computational cost and memory footprint. MobileNetV2 utilizes depthwise separable convolutions, which split the standard convolution operation into separate depthwise and pointwise convolutions, reducing the number of parameters and computations required. This architecture also introduces a novel inverted residual structure with linear bottlenecks, allowing for efficient information flow within the network. MobileNetV2 is widely used in various applications such as image classification, object detection, and semantic segmentation on resource-constrained devices.

EfficientNet B3

It is a specific variant of the EfficientNet family of convolutional neural network models. EfficientNet models are designed to achieve high accuracy while being computationally efficient in terms of both memory usage and computational resources required for training and inference.

When compared to prior versions, the EfficientNet B3 model, the third in the series, offers a middle-of-the-road level of capacity and complexity. It provides a balance between model size and performance for several computer vision applications, such as image classification, object identification, and image segmentation.

EfficientNet B3 utilizes a compound scaling method that optimizes the depth, width, and resolution of the network to achieve better accuracy and efficiency. By carefully scaling these parameters, the model can adapt to different levels of computational resources while maintaining strong performance.

In summary, EfficientNet B3 is a specific convolutional neural network architecture that balances accuracy and efficiency, making it a versatile choice for various computer vision tasks.

2.8.4 Comparison of various classifiers

This is table contain Comparison of various classifiers :

Classifier	Advantages	Drawbacks
Artificial Neural Network ANN	Faster and more accurate than KNN and MMC	Strict because the data can only belong to 1 class
Random Forest	Can classify a large data set with excellent accuracy	Constraints on storage and processing time
Multiclass-Support Vector Machine	Helps to classify the data in several classes	Not suitable when the data is noisy
Least-Square SVM	Fast and not complicated	Pruning techniques must be applied to be sparse
K-Nearest Neighbours KNN	No time spent on training	More time spent on testing and it is expensive to test each instance as well sensitive to noise and yields
Extreme learning machine ELM	Faster training and better generalization	Overfitting (occurs when a complex model has several parameters)
Naïve Bayes	Less training data is required. It works better than its counterparts when the assumption of an independent variable is true	Conditional independence may reduce accuracy
Penalized Discriminant Analysis PDA	Beneficial when the problem has a large number of noisy features	High calculation cost
Bag of Words	Uncomplicated, robust, efficient	It supposes that all words are independent of each other
CNN/Deep learning	It removes the need for a feature extraction step and classification time is shortened	A large amount of data is required for training and it is expensive to compute. They require better hardware such as Graphical Processing Unit (GPU).
Transfer Learning	This helps to apply CNN to problems with a small amount of training data	The pretrained model may not have classes with the desired labels all the time

FIG. 2.12 – Comparison of various classifiers [53]

2.8.5 DCNN[49]

DCNN is a type of DL method that differs from traditional CNN in terms of the number of hidden layers (typically more than 5), which are used to extract more features and enhance

prediction accuracy. One type of DCNN increases the number of hidden layers, while the other increases the number of nodes in the hidden layer. The DCNN method is a supervised learning task that uses raw data to identify classification features.

2.8.6 The Difference between machine learning and deep learning[54]

The difference is :

First in the fact that ML algorithms deal with quantitative and structured data and, second, the operator is responsible for choosing the right algorithm to extract the features that will influence the prediction. DL algorithms deal with unstructured data and the algorithm is trained to extract the influential elements in the prediction.

2.8.7 Image recognition technology based on deep learning[47]

The contrast between traditional image processing and deep learning methods is show in this table :

Technology	Traditional image processing methods	Deep learning methods
Essence	Manual design features + classifiers (or rules)	Automatic learning of features from large amounts of data
Method	Image segmentation method: Threshold segmentation; Roberts, Prewitt, Sobel, Laplace and Kirsh edge detection; region segmentation Feature extraction method: SIFT, HOG, LBP, shape, color and texture feature extraction method Classification method: SVM, BP, Bayesian	CNN
Required conditions	Relatively harsh imaging environment requirements, high contrast between lesion and non-lesion areas, less noise	Adequate learning data and high-performance computing units
Applicable scenarios	It is often necessary to change the threshold or redesign the algorithm when imaging environment or plant diseases and pests class changes, which has poor recognition effect in real complex natural environment	It has ability to cope with certain real and complex natural environment changes

FIG. 2.13 – Contrast between traditional image processing and deep learning methods. [47]

2.8.8 Open source tools for deep learning

It presented in this table :

Tools	Publisher	Supporting hardware	Applicable interface	Usability
Tensorflow	Google	CPU, GPU, Mobile	C, Python	Flexible development, portability, powerful performance, support for distributed applications
Torch/PyTorch	Facebook	CPU, GPU, FPGA	C, Python, Lua	Easy to debug and develop, support dynamic neural network, easy to expand, modularization and low learning cost
Caffe	BAIR	CPU, GPU	Python, Matlab	High readability, easy to expand, fast speed, large number of users and wide community
Theano	MILA	CPU, GPU	Python	Flexible and high performance

FIG. 2.14 – Comparison of open source tools for deep learning [47]

2.9 Transfer Learning (TL)[12]

TL makes use of already existing knowledge for some related task or domain in order and applies it to the problem under study. Models previously trained for image classification on large datasets are usually used and adapted to the dataset under study. A common approach is to substitute the last network layers (i.e., the dense layers) of a pre-trained network, adapting it for a different classification task. The model is then trained but only the newly inserted layers are trainable all network layers remain frozen during the training process.

In an extension of this approach, fine-tuning is also commonly used. Besides training the newly inserted layers, fine-tuning allows the training of additional layers of the base model, typically the deeper convolutional layers of the network.

TL is usually done when the studied dataset is small, with insufficient samples for training a CNN model from scratch.

2.10 Data Operations

2.10.1 Data acquisition

The data acquisition (quantity and quality) phase will have a great influence on the models both in terms of good results and robustness [12] It is the process of gathering data from various sources and systems [55]

Crop images are captured through cameras set up on the farm. Extraction of leaf images and database creation is the initial stage. Shape, texture, and leaf size are the attributes used for diagnosis [34]

Weather data collection by using IoT sensors, which typically store records.

2.10.2 Data Pre-processing[12]

It means using the information in a way that the model can easily understand [56]. In this stage, images are typically pre-processed using computer vision techniques to remove noise, enhance the image contrast, extract the regions of interest, extract image features, etc. In general, this step usually leads to better model outcomes. Here, features are normalized from vector to unit space [34]. The most common data pre-processing techniques are :

Noise Reduction

Different types of filters, such as Gaussian and median filters, are used to reduce noise to obtain smoother images. These filters have the effect of blurring and removing nonrelevant details of an image, at the expense of potentially losing relevant textures or edges [57].

Image Segmentation

It is the process of grouping pixels into regions of interest. In the context of crop disease identification, these regions for assessing the severity of the infection by the amount of the infected area, or for background removal, since the removal of the background allows highlighting the regions of interest for further analysis.

Feature Extraction

It is a common step in the pre-processing of images for shallow ML models. Different feature extractors obtain different features that can be more or less suitable for the specific problem at hand.

Cropping and Resizing Images

(a) Strawberry leaf scorch (original)   (b) Strawberry leaf scorch (segmented)

FIG. 2.15 – An example of background removal [16].

It is used for decreasing the input image dimensions, to allow greater processing speed, or to fit hardware requirements. It can also be used for creating more data to train the models [58].

Pre-Processing in Tabular Data

It consisted of weather records commonly found in the literature analyzed in the scope of this project. When gathering data records with varying dates and locations, these records can be integrated in two ways :

Cross-year where models are validated over the years at the same location.

Cross-location where models are validated across the various locations for the same year.

Common procedures in pre-processing are scaling/standardization of data and missing values processing [59]. Most algorithms require that there are no missing values in data and others, such as neural networks, can benefit from the normalization of feature values to improve training and reduce the effects of vanishing gradients [60].

Handling Null Values The easiest way to solve this problem is by dropping the rows or columns that contain null values. Sometimes, filled by the most frequent data.

Standardization We transform our values such that the mean of the values is 0 and the standard deviation is 1.Consider a data frame having 2 numerical values : Age and Weight. They are not on the same scale as Age is in years and Weight is in Kg and since Weight is more likely to greater than Age. In order to avoid this issue for each data point we just subtract the mean and divide it by standard deviation.

Normalization

It refers to the re-scaling of data features between 0 and 1, which is a special case of Min-Max scaling.

Pre-Processing in Deep Learning

It does not focus on feature extraction since one of the most essential properties of DL is its ability to generate features autonomously. For this reason, pre-processing is focused mainly on creating more images through data augmentation and resizing the input images to fit the model's input parameters.

Some studies have compared the manual selection of features with DL. When it comes to categorizing insects in the field, manually selected features were not able to capture all of the relevant information about insect infestations or to handle the noise of real-world photos. It was also not able to capture subtle differences between different insect species that share similar appearances [61].

For insect detection, DL techniques achieved higher accuracy and took less time to process since they efficiently select regions of interest [62].

When comparing the use of original color pictures with images converted to greyscale or background segmentation, DL models performed better in the original color pictures [63].

Data augmentation : is a process to artificially expand and increase the diversity of the training dataset. This process benefits the performance of the models, by introducing variability in the data and allowing a better generalization of the domain [64].. Some common transformations are rotation, cropping, scaling, and flipping.

Data cleaning : is the process of assessing the quality of the data and either modify or deleting it. It is usually applied in studies that retrieve their dataset images from search engines in an automatic way, removing pictures that do not correspond to the intended labels or that do not comply with minimum resolution requirements [65, 61].

Image resizing : is usually performed to fit the input parameters of the models. Studies have compared the performance of the models with different input image sizes and concluded that with larger images the models achieve higher accuracy but require more time for each training epoch [66]. and more powerful hardware [67].

Data Integration : To improve the accuracy and speed of the training and validation processes, the data integration technique helped us reduce and avoid redundancies in the resulting dataset [4].

Feature selection (FS)[68].

The FS model is applied to identify necessary features. The main reason to employ FS is that it enables the ML algorithm to train faster, minimizes the model complexity, and makes it easy to interpret. It also increases the system accuracy when the proper subset is selected and reduces overfitting. The computation time of the algorithm is less necessary than its classification for normal-size feature sets. But the feature selection is necessary for large datasets. Various statistical approaches can be employed in FS like a filter, embedded and wrapper methods.

Filter methods : choose the intrinsic characteristics of the features computed by univariate statistics instead of the performance of cross-validation. These methods are faster and less computationally expensive than wrapper methods.

2.10.3 Data Analysis[34].

In this step, the segmentation of images is done to find the region of interest. In segmentation, the technique used is region-based segmentation which separates healthy and diseased regions of the plant leaf by using the color of the leaf. The image acquired from the camera and the database will be pre-processed.

Next, the image resizing, followed by image enhancement and edge detection. Then many analytics techniques are carried out to classify the images according to the particular problem at hand in real-time.

The usage of the Internet of Things system makes the process easier which is extended to some level of automation. This in turn helps the user to monitor the environmental factors easily despite getting into the field physically.

2.11 Network type[47].

It can be further subdivided into classification, detection, and segmentation networks according to the different network structures.

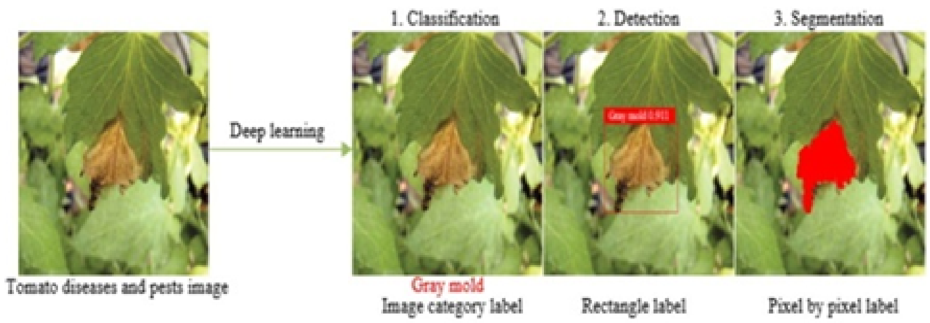


FIG. 2.16 – Definition of plant disease. [47].

2.11.1 Classification network [47].

In the real natural environment, the great differences in shape, size, texture, color, background, layout, and imaging illumination of plant diseases make recognition a difficult task. It has become the most commonly used pattern in plant disease classification. Generally, the feature extraction part of the CNN classification network consists of a cascaded convolution layer+pooling layer, followed by a full connection layer (or average pooling layer)+softmax structure for classification. It mostly uses mature network structures in computer vision, including AlexNet, GoogleLeNet, VGGNet, ResNet, Inception V4, DenseNets, MobileNet, and SqueezeNet. It works like this : inputting a test image, the network analyses and returns a label that classifies the image. It can be subdivided into three subcategories : using the network as a feature extractor, for classification directly, and lesions location.

Method	Advantages	Disadvantages
Using network as feature extractor	Obtaining effective lesion features	Relying on other classifiers for final classification results
Original image classification	Classic in structure, it is also the basis of other classification network sub-methods and can refer to many existing networks	Lesions need to account for a certain proportion in the image, otherwise their characteristics are easily pooled out, and generally only one class of lesion is allowed in an image
Classification after locating ROI	Obtaining ROI information of the lesions	Additional methods are needed to obtain ROI
Multi-category classification	Solving sample imbalance to some extent	Secondary training is needed
Sliding window	Get rough localization of lesions in images	Sliding window size requires accurate selection, and can only get rough position, slow speed of traversal and sliding
Heatmap	Generate more accurate lesion areas	Accurate lesions location depends on network classification performance
Multi-task learning network	Combining other networks to obtain exact location and category of lesions simultaneously, and reducing the number of training samples required	The network structure is relatively complex, and a pixel-by-pixel label is required when adding segmentation branches

FIG. 2.17 – Comparison of each sub-method of classification network [47].

Using network as feature extractor

First, the images are input into a pre-trained CNN network to obtain image characterization features, and the acquired features are then input into a conventional machine learning classifier (e.g., SVM) for classification.

Using network for classification directly

It is the earliest common means of CNN applied in plant disease detection. It can be further subdivided into :

1. Original image classification : That is, directly put the collected image into the network for learning and training.

2. Classification after locating ROI : we often obtain the region of interest (ROI) in advance, and then input the ROI into the network to judge the category of diseases.

3. Multi-category classification : When the number of class exceed 2 class, the network is the same as the original image classification method, that is, the output nodes of the network are the number of class+1 (including normal class). However, multi-category classification methods often use a basic network to classify lesions and normal samples, and then share feature extraction parts on the same network to modify or increase the classification branches of lesion categories. This approach is equivalent to preparing a pre-training weight parameter for subsequent multi-objective plant disease classification networks.

Using the network for lesions location

Generally, the classification network can only complete the classification of image label level. It can also achieve the location of lesions and pixel-by-pixel classification by combining different techniques and methods. It can be further divided into three forms :

1. Sliding window : This is the simplest and most intuitive method to achieve the location of the lesion coarsely. The image in this method is input into the classification network by redundant sliding on the original image through a smaller size window. Finally, all sliding windows are connected to obtain the results of the location of the lesion.
2. Heatmap : This is an image that reflects the importance of each region in the image, the darker the color represents the more important and represents the greater the probability that it is the lesion.
3. Multi-task learning network : If the pure classified network does not add any other skills, it could only realize the image level classification. Therefore, to accurately locate the location of plant diseases, the designed network should often add an extra branch, and the two branches would share the results of the feature extraction. In this way, the network generally had the classification and segmentation output of the plant diseases, forming a multi-task learning network. It takes into account the characteristics of both networks.

Confusion Matrix

A N x N matrix known as a confusion matrix is used to assess the efficacy of a classification model, where N is the total number of target classes. The matrix's actual goal values are contrasted with the model's anticipated goal values. This enables us to understand our classification model's effectiveness and the types of mistakes it produces in great detail [140].

2.11.2 Detection network[47].

Object positioning is one of the most basic tasks in the field of computer vision. It is also the closest task to plant disease detection in the traditional sense. Its purpose is to obtain accurate location and category information of the object. At present, object detection methods based on deep learning emerge endlessly. Generally, it can be divided into :

Plant diseases detection is based on two stages network :

The basic process (Faster R-CNN) is to obtain the feature map of the input image through the backbone network first, then calculate the anchor box confidence using RPN and get the proposal. Then, input the feature map of the proposal area after ROI pooling to the network, fine-tune the initial detection results, and finally get the location and classification results of the lesions.

Plant diseases detection based on one stage network

The region suggestion stage has been removed by the algorithm, which instead adds the detection head straight to the backbone network for classification and regression. This considerably accelerates the detection network's ability to draw conclusions. It is separated into two categories : SSD and YOLO, both of which use the entire picture as the network's input and give a direct output layer return of the position of the bounding box and the category to which it belongs. In contrast to the conventional CNN, the SSD chooses VGG16 as the network's trunk and incorporates a feature pyramid network to gather features from various levels and produce predictions.

The main difference between the two networks

It is that the two-stage network needs to first generate a candidate box (proposal) that may contain the lesions, and then further execute the object detection process. In contrast, the one-stage network directly uses the features extracted in the network to predict the location and class of the lesions.

Can detection network replace classification network ?

The task of the detection network is to solve the location problem of plant diseases. The task of the classification network is to judge the class of plant diseases. The detection network seems to include the steps of the classification network, that is, the detection network can answer "what kind of plant diseases and pests are in what place". But there is a misconception, about "what kind of plant diseases and pests" is given a priori, that is, what is labeled during training is not necessarily the real result. Then the involvement of the classification network is necessary. Thus, the detection network cannot replace the classification network.

2.11.3 Segmentation network[47]

It converts the detection task to semantic and even instance segmentation of lesions and normal areas. It not only finely divides the lesion area, but also obtains the location, category, and corresponding geometric properties (including length, width, area, outline, center, etc.). It can be roughly divided into Full convolution neural network (FCN), and Mask R CNN.

Compared with the classification and detection network methods, the segmentation method has advantages in obtaining lesion information. However, like the detection network, it requires a lot of annotation data, and its annotation information is pixel by pixel, which often takes a lot of effort and cost.

2.12 Evaluation indices[47].

It can vary depending on the focus of the study. It is used to evaluate the performance of algorithms. Common evaluation indices presented in this table :

	Condition (e.g. disease detection)		
	True (inoculated leaf)	**False** (healthy leaf)	
Test result	No. of true positive (TP)	No. of false positive (FP)	Pos. predictive value (Precision) ($TP\,(TP+FP)$)
Test result	No. of false negative (FN)	No. of true negative (TN)	Neg. predictive value ($TN\,(TN+FN)$)
	Sensitivity (Recall) $\frac{TP}{TP+FN}$	**Specificity** $\frac{TN}{TN+FP}$	**Accuracy** $\frac{TP+TN}{TP+FP+TN+FN}$

FIG. 2.18 – Important measures in a statistical classification task [101].

2.13 Challenges[47].

2.13.1 Small dataset size problem

self-collected data sets are small in size and laborious in labeling data. Compared with more than 14 million sample data in ImageNet datasets, the most critical problem facing plant disease detection is the problem of small samples. In practice, some plant diseases have low incidence and high cost of disease image acquisition, resulting in only a few or dozen training data collected, which limits the application of deep learning methods. In fact, for the problem of small samples, there are currently three different solutions :

Data amplification, synthesis and generation

Data amplification is a key component of training deep learning models. An optimized data amplification strategy can effectively improve the plant disease detection effect. The most common method of plant disease image expansion is to acquire more samples using image processing operations such as mirroring, rotating, shifting, warping, filtering, contrast adjustment, and so on for the original plant disease samples. In addition, Generative Adversarial Networks (GANs) [70]. and Variational automatic encoders (VAE) [69]. can generate more diverse samples to enrich limited datasets.

Reasonable network structure design : By designing a reasonable network structure, the sample requirements can be greatly reduced.

2.13.2 Detection performance under the influence of illumination and occlusion

Lighting problems

Natural light changes very dynamically, and the range in which the camera can accept dynamic light sources is limited, it is easy to cause image color distortion when above or below this limit. In addition, due to the difference of in view angle and distance during the image collection, the apparent characteristics of plant diseases and pests change greatly, which brings great difficulties to the visual recognition algorithm.

Occlusion problem

They are common in real natural environments, including blade occlusion caused by changes in blade posture, branch occlusion, light occlusion caused by external lighting, and mixed occlusion caused by different types of occlusion. The difficulties of plant disease identification under occlusion are the lack of features and noise overlap caused by occlusion. Different occlusion conditions have different degrees of impact on the recognition algorithm, resulting in false detection or even missed detection.

Detection speed problem

Compared with traditional methods, DL algorithms have better results, but their computational complexity is also higher. If the detection accuracy is guaranteed, the model needs to fully learn the characteristics of the image and increase the computational load, which will inevitably lead to slow detection speed and cannot meet the needs of real-time.

To ensure the detection speed, it is usually necessary to reduce the amount of calculation. However, this will cause insufficient training and result in false or missed detection. Therefore, it is important to design an efficient algorithm with both detection accuracy and detection speed.

Plant disease detection methods based on deep learning include three main links in agricultural applications : data labeling, model training, and model inference [71].

2.14 Computer vision[72]

It is a branch of artificial intelligence that enables computers and systems to gather valuable data from digital images, videos, and other visual inputs and take actions or offer suggestions based on that data. Computer vision offers machines the ability to perceive, observe, and understand, just like artificial intelligence gives them the capacity to think.It works very much like how the human eye does.

2.15 Precision agriculture[73]

The goal behind precision agriculture (PA) is to monitor, measure, and respond to crop variability both within and between fields. PA is also known as as-needed farming, satellite agriculture, precision farming, and site-specific crop management (SSCM).

To ensure that soil and crops receive the precise nutrients they need for optimum health and productivity, precision agriculture uses information technology (IT). By doing this, sustainability, profitability, and environmental protection are all guaranteed. It considers elements including soil type, topography, weather, plant growth, and yield statistics while managing crops.

2.15.1 The benefits of precision agriculture[73]

Predictive analytics software makes use of the data after it has been collected to provide farmers advice on crop rotation, and the ideal times to sow, harvest, and maintain the land.

Agricultural control centers may combine sensor data and imaging input with other data to assist farmers in identifying crops that require treatment and determining the appropriate amount of water, fertilizer, and pesticides to administer.

In addition to ensuring that the soil has the ideal balance of additives for maximum health, this aids the farmer in preventing resource waste and run-off, cutting expenses, and managing the farm's environmental effects.

But today, farming cooperatives and even tiny family farms may practice precision agriculture thanks to smartphone applications, intelligent sensors, drones, and cloud computing.

2.15.2 Are there any challenges involved with precision agriculture ?[73] ?

- There are difficulties with precision agriculture. The integration of all the many data sources is one of the largest hurdles.
- It necessitates a large technological investment.
- It may be expensive to purchase the hardware and software needed for precision agriculture, and it takes time to become proficient with it all.

2.15.3 The use cases for precision agriculture[73]

Some of the most popular applications for precision agriculture today include :

- Agricultural mapping and field scouting. Drones equipped with cameras can create high-resolution maps of fields.
- Soil sampling and analysis. Mobile apps can collect data about soil type, fertility, moisture content, and more.
- Weather monitoring. Hyperlocal weather data can help users decide when to plant, how much water to give crops and when to harvest.

- Labor management. GPS-enabled mobile apps can track the location and activity of workers in the field.
- Equipment management. It can help farmers keep track of their equipment, schedule maintenance and plan for repairs.

2.16 Smart Farming (SF)[74]

It is the application of advanced technology like AI, IoT, etc. to make crop production more advanced. By using SF techniques, farmers can boost production and minimize the wastage of crops. These technologies help farmers in various stages, like plantation, irrigation, soil quality checking, weather reporting, etc. in crop production.

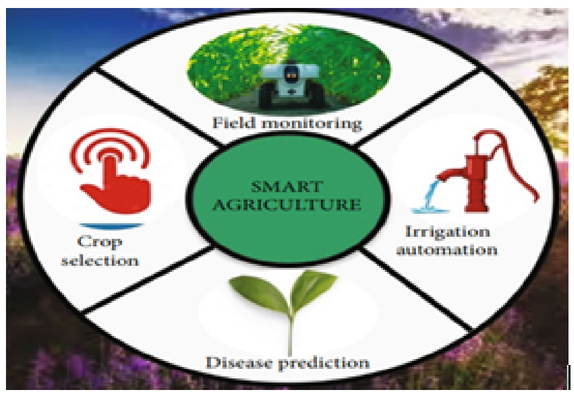


FIG. 2.19 – Sample illustration of smart farming. [75]

2.17 Internet of Things (IoT)

It is a network of interconnected devices which communicate sense and interact with each other. Adaptation of IoT in various fields has been increasing rapidly, even though the technology is relatively not new. These devices help build a network between them and are capable of sharing information and data, as well as actively acting on instructions given to them [74]. IoT offers appropriate solutions for multiple applications such as smart health care, smart cities, security, retail, traffic congestion industrial control, and agriculture [76]. IoT has brought a great revolution in the agricultural environment by examining multiple complications and challenges in farming [77]

2.17.1 Major components of IoT based Smart Farming[78]

It consists of four major components :

The physical structure : is the most important factor for precision agriculture to avoid any unwanted happening. The whole system is designed in such a way that controls the sensors, actuators, and devices. A sensor performs multiple tasks like soil sensing, temperature sensing, weather sensing, light sensing, and moisture sensing. Similarly, devices perform many control functions like node discovery, device identification, naming services, etc. All these functions are performed by any device or sensor which is controlled through a microcontroller. This controlling operation is performed by any remote device or computer that is connected to the Internet.

Data Acquisition : is further divided into two sub-components namely : IoT data acquisition and standard data acquisition. Whereby,

1 IoT data acquisition : component consists of seven protocols that are Message Queuing Telemetry Transport (MQTT), Websocket, Advanced Message Queuing Protocol (AMQP), Node, Constrained Application Protocol (CoAP), Data Distribution Service (DDS), and Hyper Text Transfer Protocol (HTTP). Depending on the requirements and conditions more protocols can be used for the implementation of smart farming.

2 Standard data acquisition : ZigBee, WIFI, Long Range Wide Area Network (LoraWan), SigFox and ISOBUS protocols have been used.

Data processing : consists of multiple features that are image or video processing, data loading, decision support system, and data mining. According to the system requirements, any feature may be added that may work in parallel to provide other services.

Data analytics : consists of two main features which are monitoring and controlling. Monitoring involves three main applications in a smart agriculture that are Live Stock Monitoring, Field Monitoring, and Greenhouse Monitoring.

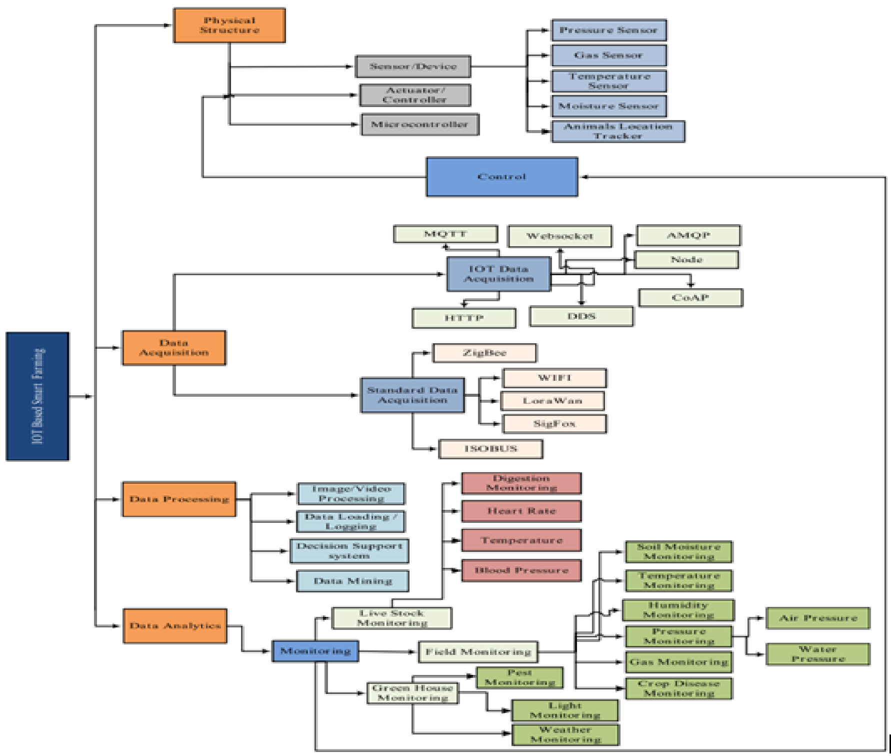


FIG. 2.20 – Major components of IoT based smart farming [78]

2.17.2 IoT Agriculture network architecture[78]

Most the IoT applications usually follow the four-layer architecture :

APPLICATION LAYER

Due to the energy constraints and stringent computation involved by the IoT devices. These protocols can be increased or decreased according to the system requirement ; there are many lightweight protocols on the application layer such as :

1 CoAP protocol : it runs on UDP and works on the principle of request or response architecture [79]

2 AMQP protocol : it runs over the TCP protocol by following publish/subscribe architecture asynchronously and uses TSL/SSL for security assurance.

3 MQTT : it is a bandwidth-efficient protocol that uses little battery power and is designed for receiving and transmitting sensor information [80]

4 HTTP : it is a well-known web messaging protocol based on the request/response architecture [81]. Runs over TCP and does not define any QoS, uses TSL/SSL for security purposes.

TRANSPORT LAYER

This layer is also called the host to the host transport layer and is directly transferred from the IP to the IoT domain. The main task of the network layer is to collect and encapsulate the agricultural information which is obtained through the sensor layer. Two protocols are :

1 Transmission control protocol (TCP) : it is a connection-oriented protocol that ensures the reliability of delivered data. TCP data transmission speed is low as compared to UDP.

2 User datagram protocol (UDP) : it is a connectionless protocol that does not ensure the reliability of data. Its data transmission speed is high as compared to TCP. Both of these protocols are used in different applications because their choices depend upon the requirements of the application.

NETWORK LAYER

This layer is an indispensable technology for precision farming and is responsible to transmit agricultural information at the application layer. IP is the major choice with the existing two versions that are IPv4 and IPv6. IPv4 came into existence due to increasing the large number of addressable devices. Whereas, the invention of IPv6 was expected which gradually establish on all networking devices.

ADAPTATION LAYER (AL)

AL's aim is to ensure interoperability and implement fragmentation, compression, and reassembly mechanism. Although AL attained many advances but still there is a complexity for IPv6 support because its direct use on IoT devices is not considered reasonable.

PHYSICAL AND MAC LAYERS

This is the bottommost layer in the agriculture network architecture which is responsible to sense and actuate different agricultural parameters. Within the physical and MAC layer IEEE 802.15.4 is one of the most popular standards which was designed for low cost, low consumption, and low complexity [82]

2.17.3 The benefits of IoT[83]

Some of the common benefits of IoT are :

- Monitor their overall business processes.
- Improve the customer experience (CX).
- Save time and money.
- Enhance employee productivity.
- Integrate and adapt business models.
- Make better business decisions.
- Generate more revenue.
- Using sensors and other IoT devices, manufacturing, transportation, and utility companies employ IoT the most frequently.
- The availability of information on any device, at any time, from anywhere.
- Better communication between electronic devices that are connected.
- Saving time and money by sending data packets through a network connection ; automating processes to raise the standard of a company's services and lessen the need for human involvement.

2.17.4 Disadvantages of IoT[83]

IoT has a number of drawbacks, including the following :

• As there are more devices that are linked and more information is shared between them, there is a greater chance that a hacker would steal sensitive data.

• Enterprises may someday have to deal with enormous numbers of IoT devices—possibly millions—and it will be difficult to collect and manage the data from all of those devices.

• Since there is no global IoT compatibility standard, it is challenging for devices from various manufacturers to communicate with one another. • If there is a problem in the system, it is possible that every linked device would become corrupted.

2.18 Smart Agriculture Monitoring (SAM)

IoT can provide automation in agriculture monitoring systems with the help of sensors, actuators embedded hardware platforms, and cloud-enabled technologies like big data and cloud computing. Generally, the IoT role in agriculture has been transformed [84] into different domains such as water and soil management, crop monitoring, etc. WSN also enables automation in the field of agriculture with energy harvested [85] nodes, hardware cost-effectiveness, and scalability.

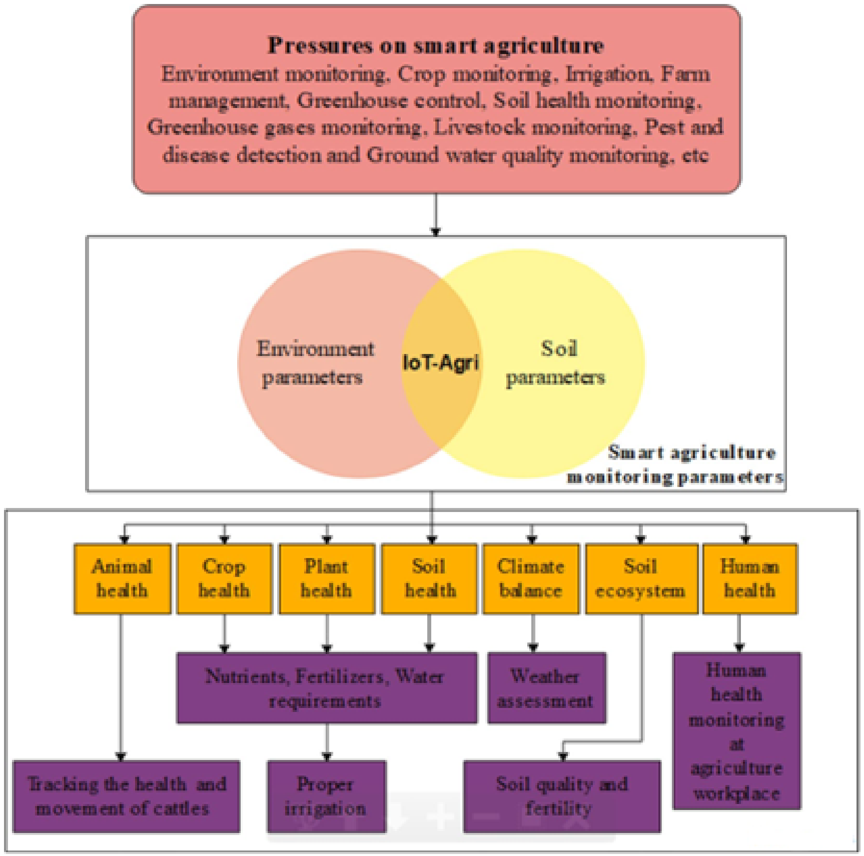


FIG. 2.21 – Overview of the smart agriculture parameters. [85]

2.19 Overview of Wireless Communication Technologies

In many of the applications, WCT has been used. Based on the transmission distance the wireless technologies are divided into 3 categories :

They are short-range communication (SRC) [88] : for a distance of <10 m. It includes RFID(Radio-frequency identification), BLE (Bluetooth), UWB (ultra-wideband), and so on.

Medium-range communication (MRC) : for the distance of (10–100) m. It includes Wi-Fi and ZigBee technology [73]

Long-range communication (LRC) : for distance> 100 m. In LRC, cellular networks are used, like 2G/3G/4G [57]

IoT protocols for agriculture :

This table represents the IoT protocols :

Parameters	*Standard*	*Frequency Band*	*Data Rate*	*Transmission Range*	*Energy Consumption*	*Cost*
WiFi	IEEE 802.11 a/c/b/d/g/n	5 GHz–60 GHz	1 Mb/s– 7 Gb/s	20–100 m	High	High
LoRaWAN	LoRaWAN R1.0	868/900 MHz	0.3-50 Kb/s	<30 KM	Very Low	High
WiMAX	IEEE 802.16	2 GHz–66 GHz	1 Mb/s–1 Gb/s (Fixed) 50–100 Mb/s (mobile)	<50 Km	Medium	High
Mobile Communication	2G-GSM, CDMA.- 3GUMTS, CDMA2000, 4G-LTE	865 MHz, 2.4 GHz	2G: 50–100 kb/s 3G:200 kb/s 4G:0.1–1 Gb/s	Entire Cellular Area	Medium	Medium
LR-WPAN	IEEE 802.15.4	868/915 MHz, 2.4 GHz	40–250 Kb/s	10–20 m	Low	Low
RFID	ISO 18000-6C	860-960MHz	40 to 160 kbit/s	1-5 m	Low	Low
ZigBee	IEEE 802.15.4	2.4 GHz	20-250 Kb/s	10–20 m	Low	Low
MQTT	OASIS	2.4 GHz	250 kbps	-	Low	Low
SigFox	SigFox	200 kHz	100-600 bit/s	30-50 km	Low	Low
Bluetooth	IEEE 802.15.1	24GHz	1-24 Mb/s	8-10m	Very Low	Low

Fig. 2.22 – Comparison of existing wireless protocols [78]

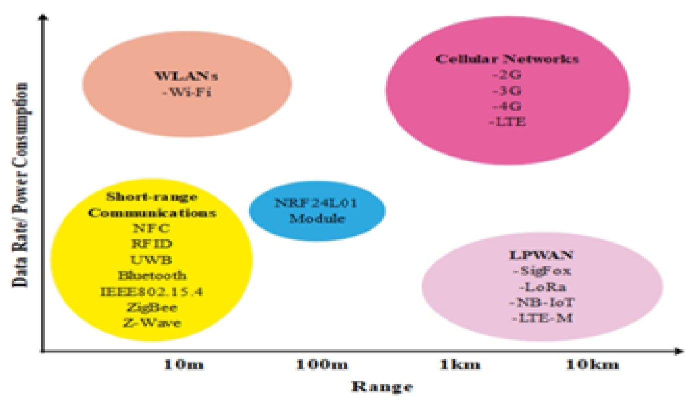


Fig. 2.23 – Various wireless communication technologies for range and data rate comparisons [86]

2.19.1 Wireless sensors (WS)[8]

WS is the most crucial and plays a key role when it comes to collecting crop conditions and other information. In the following, major sensor types are discussed according to their

working procedure and purpose and the benefits they offer :

Filed-programmable gate array(FPGA)-Based sensors

FPGA-based sensors are starting to be used in agriculture recently due to their flexibility of reconfiguration. The major options where these can be employed include measuring real-time plant transpiration, irrigation, and humidity [89, 90]. However, their utilization in agriculture is in the early stages due to their limitations, such as size, cost, and power consumption.[8]

Optical sensors

These sensors use light reflectance phenomena and help to measure soil organic substances, soil moisture, and color, the presence of minerals and their composition, clay content, etc. [91, 92]

Airflow sensors

These sensors are capable of measuring soil air permeability and percentage of moisture and identifying soil structure to distinguish different types of soils [93]

Electochemical sensors

These are mostly used to assess the significant soil characteristics to analyze the soil nutrient levels, such as pH [94]

Softwar level-based(SWLB) sensors

They are being utilized in agriculture catchments to characterize hydrological behaviors, such as water level and flow, at adjustable time-step acquisitions. This is done by measuring rainfalls, stream flows, and other water presence options [95, 96]

Telematics sensors

They support telecommunication between two places—more precisely, among two vehicles when considering agriculture-based applications. They are used to collect data from remote locations (especially inaccessible points), operations of machines that report on how the components are working, and record location and travel routes to avoid visiting the same patch [97]

Remote sensing

They are used to capture and store geographic information, and further analyze, manipulate, manage, and present all types of spatial or geographical data [8]

2.20 Cloud Computing [98]

It entails the delivery of a range of services over the Internet. These applications and tools comprise servers, databases, networking, and data storage, among other things.

Cloud-based storage enables you to store files in a remote database as opposed to a local storage device or a customized hard drive. As long as a device has internet connectivity, it can obtain the data and software it needs to function.

It is a popular option for both individuals and businesses because of a number of advantages, including cost savings, increased productivity, speed and efficiency, performance, and security.

2.20.1 Cloud computing categories[98]

It does not consist of a single technological component like a microprocessor or a telephone. Instead, it is a system made up mostly of three services :

Software-as-a-service (SaaS) : A software program is licensed to clients as part of the SaaS model. Usually, licenses are made available on-demand or on a pay-as-you-go arrangement. Microsoft Office 365 contains a mechanism like this.

Infrastructure-as-a-service (IaaS) : is a technique for offering anything over an IP-based connection as part of an on-demand service, from operating systems to servers and storage. Clients can obtain software and servers through an on-demand, outsourced service rather than having to buy them outright. IaaS systems like IBM Cloud and Microsoft Azure are well-known examples.

Platform-as-a-service (PaaS) : is regarded as cloud computing's third and most challenging layer. The key difference between PaaS and SaaS is that PaaS is a platform for building software that is given through the Internet as opposed to providing software as a service. This tactic makes use of tools like Heroku and Salesforce.com.

2.20.2 Deployment Models[98]

Public clouds : use Internet-based storage and computers to deliver their services. These are run by independent firms that manage and take care of all the infrastructure, software, and hardware. Customers use accounts that virtually anybody may use to obtain services.

Private clouds : are only accessible to selected clients, often one company or organization. The cloud computing service could be hosted by the company's data center. On a private network, several private cloud computing services are offered.

As the name suggests, **hybrid clouds** combine both public and private services. This kind of architecture gives the user more options and improves the infrastructure and security for the user.

2.20.3 Advantages of Cloud Computing[98]

• Cloud-based software has several advantages for businesses across various industries, including accessibility from any device through a browser or native app.

• It involves much more than simply accessing data across many devices. Large firms may save a ton of money this way as well.

• People may conserve storage space on their computers or laptops by using the cloud infrastructure.

2.20.4 Disadvantages of the Cloud [98]

There are hazards, of course, with all the speed, efficiency, and innovations that come with cloud computing.

• Security has always been a major worry with the cloud, particularly when it comes to private financial and medical documents.

• Power outages, internal issues, and natural calamities can also affect servers operated by cloud computing organizations.

• There is a learning curve for both employees and management, as with any technology. However, since so many people may access and alter data through a single gateway, unintentional errors might spread across the whole system.

2.21 AGRICULTURAL DRONES

Drones are defined as Unmanned Aerial Vehicles (UAVs) which are being utilized in agriculture to improve various practices of farming. These flying devices are controlled remotely by remote control or autonomously programmed. Agricultural processes which are performed by drones are crop health assessment, spraying, screening, planting, scouting reports, measurement of nitrogen in wheat, and analysis of soil conditions. Drones facilitate the farmers via integration with Geographic Information Systems (GIS) mapping, and crop health imaging. Drones are mostly deployed in large farms. In the area of agriculture pesticides and fertilizers are very important for crop yield [99, 100]. Drones are high-speed and effective in the spraying operation.

2.22 Conclusion

In conclusion, the prevention, detection, and control of agricultural illnesses may be radically altered by the use of artificial intelligence (AI) and the Internet of Things (IoT) to the management of plant diseases. Real-time monitoring of environmental variables, early illness identification, and precise therapies are made possible by AI algorithms working in conjunction with IoT sensors. Farmers now have access to precise disease risk assessments and useful insights because to the capacity to evaluate vast amounts of data, such as satellite images, weather patterns, and historical disease records.

AI-powered image analysis can quickly and accurately diagnose plant diseases, aiding in timely treatment and reducing yield losses. Additionally, predictive models based on AI can forecast disease outbreaks, providing early warnings to farmers and enabling proactive disease management strategies.

The integration of IoT devices and AI algorithms allows for automated and targeted interventions. Smart sensors can monitor soil moisture, temperature, and other variables, optimizing irrigation practices and reducing the risk of waterborne diseases. Drones equipped with AI algorithms can survey large fields, identifying disease hotspots and enabling targeted pesticide applications.

The combination of AI and IoT offers the potential to optimize resource usage, reduce chemical inputs, and minimize the environmental impact of disease management practices. By providing real-time insights, these technologies empower farmers to make informed decisions, enhancing productivity and sustainability in agriculture.

However, successful implementation requires addressing challenges such as data privacy, interoperability, and providing training and support to farmers. Collaboration among researchers, farmers, technology developers, and policymakers is crucial to ensure the effective integration of AI and IoT into plant disease management.

Overall, the combination of AI and IoT holds great promise for revolutionizing plant disease management. These technologies offer increased precision, early detection, and proactive strategies, leading to improved crop health, higher yields, and more sustainable agricultural practices. By harnessing the power of AI and IoT, we can enhance global food security and build resilient agricultural systems in the face of plant diseases.

Chapter 3

PHASE-3- REALIZATION AND IMPLEMENTATION

3.1 Introduction

The implementation of machine learning models has revolutionized several aspects of agriculture, including plant disease prediction, crop recommendation, weed detection, pest detection, and fertilizer optimization. These applications leverage the power of artificial intelligence to enhance crop yield, reduce losses, and improve overall agricultural practices. In this introduction, we will provide an overview of the implementation of models for each of these areas.

Plant Disease Prognosis : Crop productivity and quality can be severely impacted by plant diseases. Machine learning algorithms may reliably anticipate and diagnose possible diseases by analyzing a variety of variables, including environmental circumstances, symptom patterns, and disease development. These models can provide early warnings to farmers, allowing them to take preventive action and administer tailored treatments, lowering crop losses. They do this by leveraging historical data and real-time monitoring.

Crop Recommendation :

Choosing the right crop for a specific location and environmental conditions is crucial for maximizing yield. Machine learning models can analyze diverse datasets, including soil quality, climate data, historical yield records, and market demand to recommend the most suitable crop for a given area. These models consider various factors such as temperature, rainfall, soil composition, and market trends to provide personalized recommendations, allowing farmers to make informed decisions and optimize their agricultural practices.

Weed Detection :

Weeds compete with crops for resources, affecting their growth and yield. Machine learning models can analyze images captured by drones, satellites, or on-field sensors to identify and classify different weed species. These models employ computer vision techniques and deep learning algorithms to detect the presence of weeds accurately. By identifying weed-infested areas, farmers can apply targeted herbicide treatments, reducing chemical usage and minimizing crop damage.

Early detection of pests is essential for successful pest control since they constitute a serious danger to agricultural output. To forecast the risk of pest infestations, machine learning algorithms may assess a variety of data sources, including meteorological conditions, insect activity patterns, and crop attributes. These models may pinpoint individual pest species, evaluate their population patterns, and offer insights on when to use pest management strategies most effectively. Farmers can limit crop losses and the effects of pests by permitting prompt response.

Fertilizer Optimization :

Efficient use of fertilizers is essential to promote crop growth while minimizing environmental impact and production costs. Machine learning models can analyze soil nutrient data, historical yield records, and crop characteristics to recommend precise fertilizer dosages for different fields. These models consider factors such as soil composition, nutrient requirements, and crop growth stages to optimize fertilizer application. By providing tailored fertilizer recommendations, farmers can enhance nutrient utilization, reduce fertilizer waste, and improve overall sustainability.

In conclusion, the implementation of machine learning models for plant disease prediction, crop recommendation, weed detection, pest detection, and fertilizer optimization has transformed agricultural practices. These models empower farmers with valuable insights, enabling them to make informed decisions, minimize losses, and optimize resource utilization. By leveraging the power of artificial intelligence, the agriculture industry can strive towards sustainable and efficient food production systems.

3.2 Description of the prototype

We want to develop an IoT-based monitoring system for precision agriculture applications such as plant disease control. Such an agricultural monitoring system provides environmental monitoring services that maintain the crop growing environment in an optimal status and early predict the conditions that lead to plant disease outbreaks. This system provides a service to store the environmental and soil information collected from a wireless sensor network installed in the planted area in a database. Furthermore, it allows users to monitor environmental information about the planted crops in real time through any Internet-enabled devices. We try to develop artificial intelligence prediction algorithms to realize a system that allows emulating the decision-making ability of a human expert.

3.3 Conception of the system

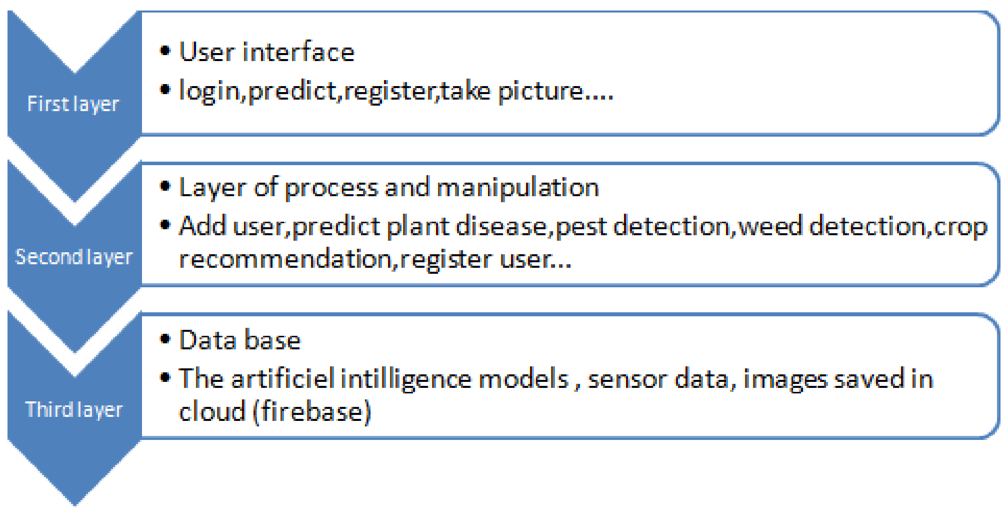


FIG. 3.1 – General conception of my system.

And this architecture represents how system work :

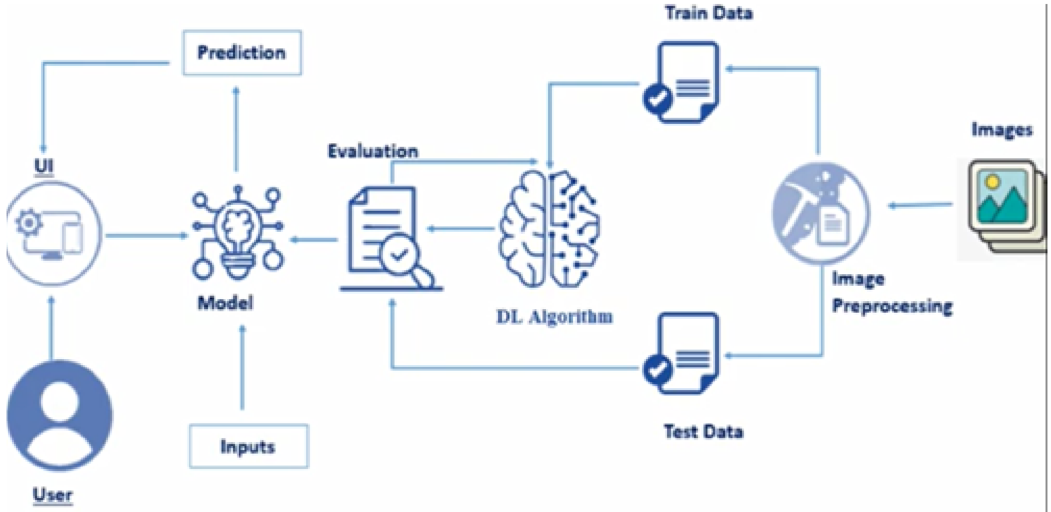


FIG. 3.2 – Architecture represents how system work. [98]

3.4 Materials and methods

The proposed monitoring system for early plant disease is composed of two main modules :

- **The hardware module :** is mainly responsible for collecting information from the outdoor environment, then transmitting the collected data using a communication subsystem.

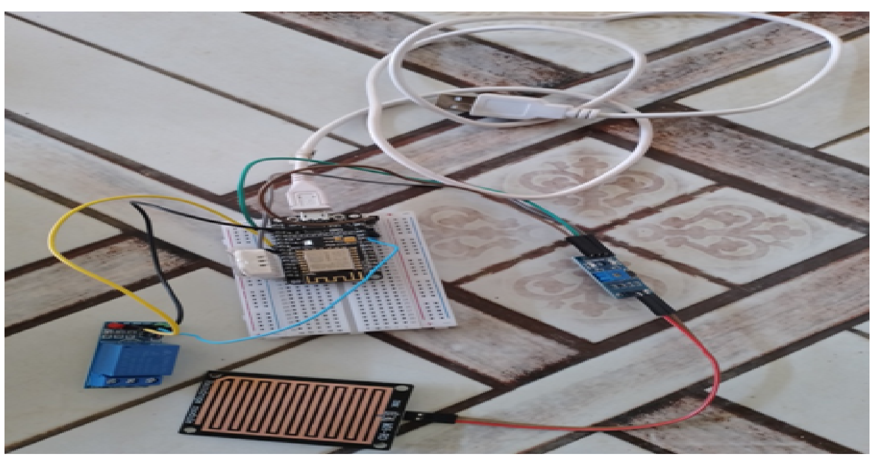


FIG. 3.3 – Architecture of Hardwar component of AI- AgriPlantD.

- **The software module :** has four main functions :
- It collects, processes, stores, and presents the data provided by the sensors.
- It provides a user-friendly interface to the system.
- It predicts plant diseases.
- It suggests suitable preventive actions (applying pesticides, for instance) by

analyzing the collected data in light of the disease models.

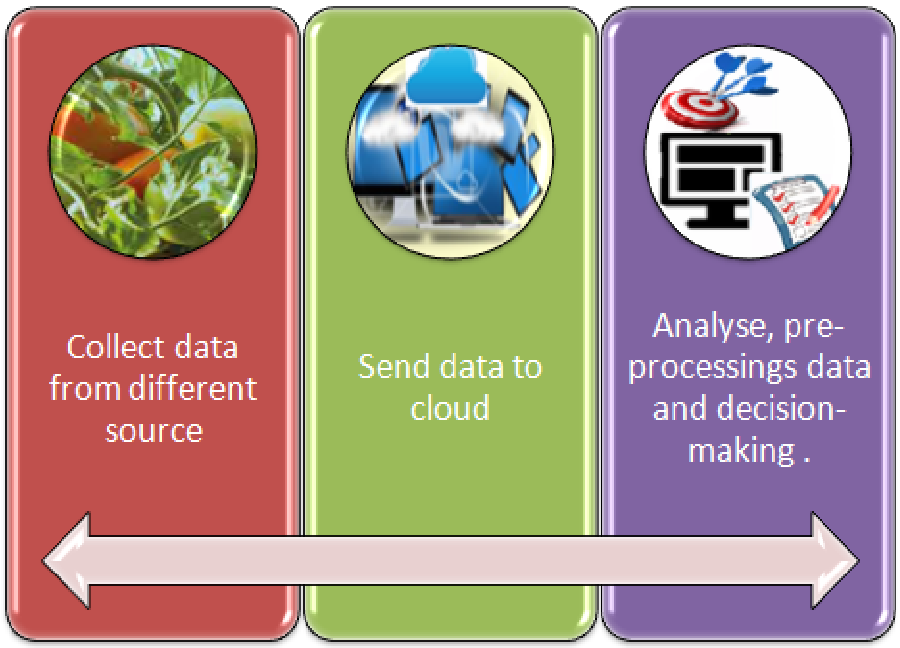


FIG. 3.4 – General architecture of AI- AgriPlantD.

3.5 How systems work ?

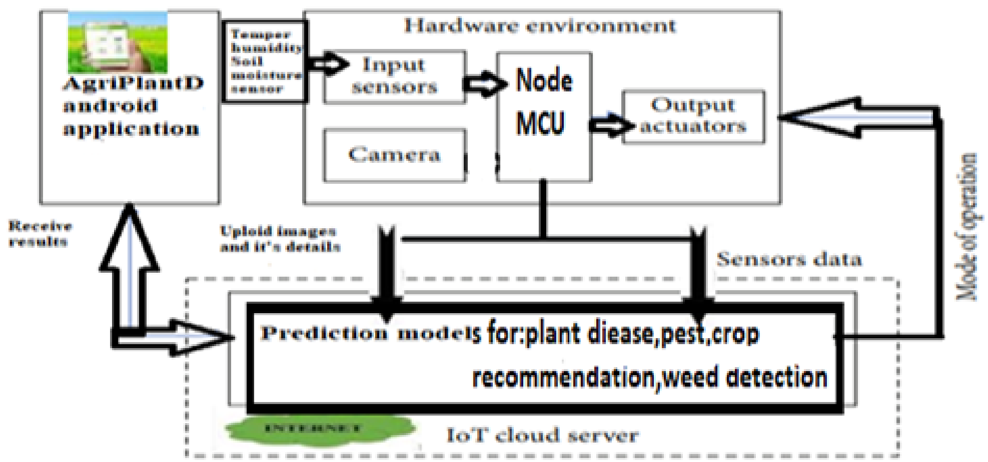


Fig. 3.5 – Proposed architecture of AI- AgriPlantD with IoT.

This section provides the implementation details of AI- AgriPlantD. AI- AgriPlantD is developed with a user-friendly environment for farmers using wifi Node MCU and IoT environment with AgriPlantD application. Farmers monitor and control their farm fields with AgriPlantD. The instrumentation system is placed in the farm field and monitors plant statistics using various sensors. Moreover, all these sensor data are uploaded in the cloud-based IoT environment. An artificial intelligence system is placed via the cloud served with CNN, which continuously monitors sensor data and plant disease status and sends necessary alerts to farmers using the application. Finally, nutrients are supplied to the plants according to the levels stated by the grower. In addition, nutrients are applied to plants with standard reference levels.

Moreover, the sensor values are continuously updated in the IoT-based cloud environment. Node MCU receives all sensor values and sends this data to a cloud server. Here, the ANN-Forecasting model is used effectively to determine the type of disease based on the temperature and humidity of the air, the value of the rain sensor, and the soil moisture. Therefore, it is necessary to identify plant diseases at an early stage. Therefore, the CNN prediction model is used to identify the different types of plant diseases from the images captured by the smartphone camera. In addition, the values update for the farms was done through the mobile application.

The application is developed with a login page. Therefore, hackers cannot control the domain and cannot access the application. After logging in successfully, the farmer can monitor and control the field using the 'Plant Disease Prediction', 'Farm Sensor Data', and 'Farm Control' buttons. On the Farm Sensor Data page, different types of sensor data (air temperature, humidity levels, nitrogen, phosphorus, potassium...).

The readings are compared to the desired values as input by the user. If the readings are good under the border conditions, the plant is declared disease-free. But if you feel the plant's temperature is higher than desirable, it may be infected. Likewise, if a plant is not the color it should be, it may have an infection. This is detected by the light sensor. Finally, if the plant has a higher moisture content value, the plant may decompose. but

If the plant has too low humidity, the automatic watering system is triggered to water the fields. This model will help in providing crop forecasts and will allow the farmer to take appropriate actions.[104]

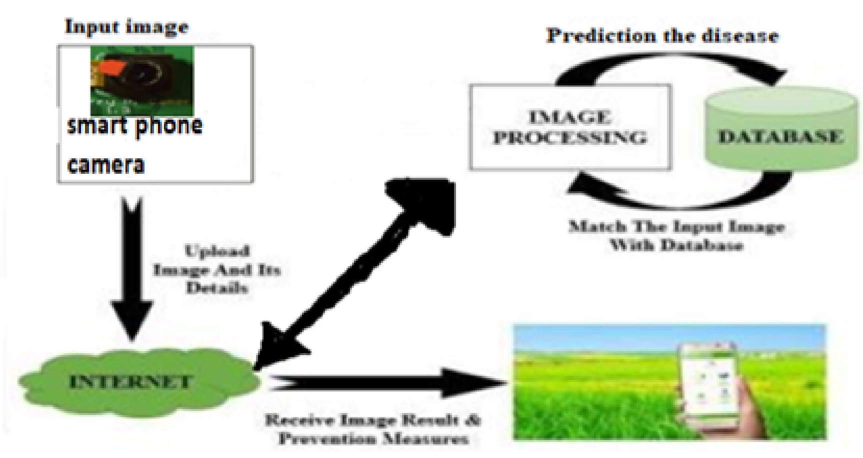


FIG. 3.6 – Architecture of operation of prediction disease by image.

3.6 Development of Experimental Setup[86]

In this section, the description of the proposed system, which helps in the development of agricultural applications, with the help of different required hardware components,[105] for making the system efficient and intelligent. The hardware components are classified into five types :

3.6.1 Field Sensors

In various applications of agriculture, the environment sensors (DHT22) and deployment sensors (soil temperature and soil moisture, ph sensor) play a very significant role.

3.6.2 Microcontroller

In this view, an implementation of the proposed architecture consists hardware microcontroller board (Raspberry Pi).

3.6.3 Wireless Communication Technologies and Protocols

In this study, WCT and protocols are used for data transmission for the range of short and long distances. IoT is having many WCT as well as protocols. Apart from these, some of them have been discussed for sending the data to a cloud server like wifi module Node MCU and as well as protocols like MQTT, CoAP for providing communication between the machine to machine (M2M) and messages are sent to the cloud.

3.6.4 Field Actuators

In the field of agriculture applications, field actuators have provided the controlling functions of devices. In this scenario, field actuators are like the Relay module and water pomp.

3.7 Description Architecture IoT used[86]

The proposed architecture is a self-powered, real-time, IoT based-cloud enabled service for smart agriculture, decision-making system consists of two sections :

3.7.1 The transmitter section (TX)

It is responsible for monitoring the agriculture field to get data from sensors. So the physical layer comes into the picture and it senses all agriculture parameters through sensors like temperature, humidity, moisture, and so on.

It sends the digital signal to the above level, i.e., the conceptual layer. Data passing tasks received from the conceptual layer are handled by the communication layer.

The Internet layer is the most among the three layers which form the system backbone. ThingSpeak cloud works in the internet layer.

The application layer supports the network of IoT development on a large scale.

All the applications, services, and research activities connected to a network of IoT have come under the business layer.

Collect data from sensors. Once the completion of the data has been completed, then it gets processed and this processed data is called data processing which comes under hardware embedded platforms (IoT gateway), along with WCT for IoT, based on the conceptual and communication layers. Through lightweight communication protocols like MQTT and CoAP, the messages are transmitted from client to server based on a set of rules for data in formats such as XML, JSON, CSV, and so forth. These things will happen while the Internet is available at the farmer's place and it comes under the Internet layer.

3.7.2 The receiver section (RX)

It aims to take care of the agriculture field information which is stored in the cloud server repository (IoT-cloud) with the help of IoT security like API key (Application Programming Interface), so that data gets access from the cloud repository through preconfigured devices like mobile phones, laptops and so forth.

The application layer is responsible to access the data from the cloud through a farmer's mobile phone regarding agriculture data. It is built in order to undergo complete automatic visualization through graphical representation, monitoring for real-time applications, and statistical analysis for data. Data storage and data analysis is done in IoT cloud itself.

Finally, IoT cloud platforms are providing cloud services for data storage and analysis. These cloud platforms are associated with architecture, of IoT for providing cloud services such as Software-as-a-Service (SaaS), Platforms-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS).

The wireless connection is established between AWMU (TX) to WAMU (RX)

Agriculture Wireless Monitoring Unit (AWMU)

AWMU is deployed in the field of agriculture. It consists of environmental sensors, agricultural sensors, a hardware platform (Raspberry Pi), an energy harvesting capability circuit for power supply, and a wireless transmitter module (NRF24L01). The sensors like DHT22 (temperature and humidity sensor), soil moisture sensors, and NRF24L01-TX module are connected to the microcontroller. The AWMU transmits information to other devices through wireless communication, over a network with good internet connectivity. In the agriculture field, each AWMU is communicated through Wi-Fi connectivity. Finally, the field information is sent through the IoT gateway from AWMU to the receiver module.

Wireless Actuator Monitoring Unit (WAMU)

WAMU is deployed in the field of agriculture. It has four modules such as :

- Relay module
- Motor module

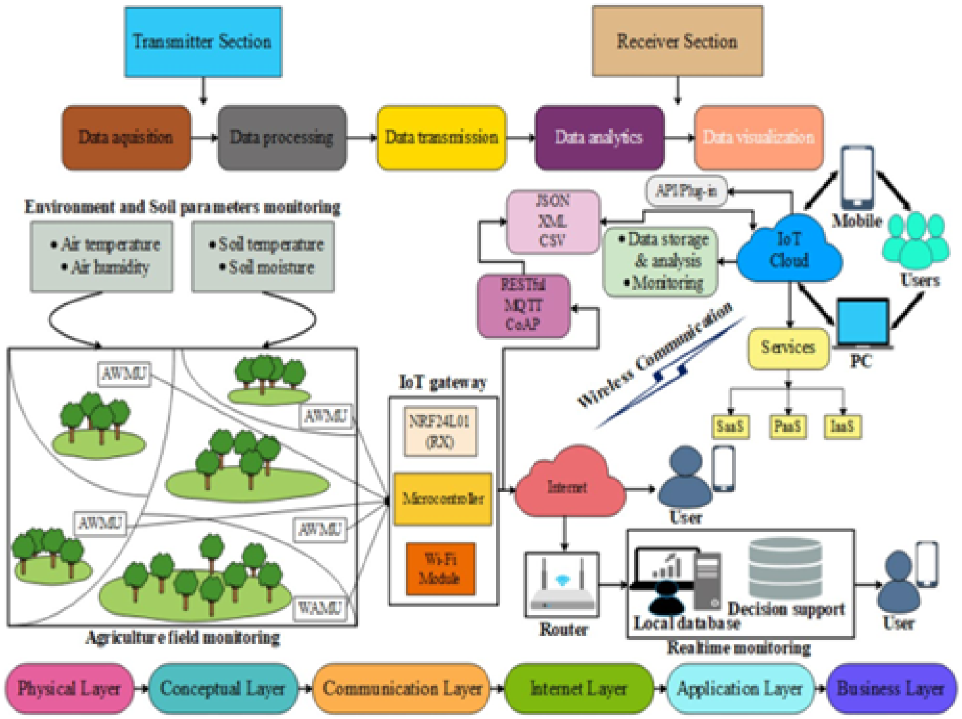


Fig. 3.7 – Proposed architecture for the smart agriculture decision-making system [86]

3.7.3 Field Estimation for Deployment Sensors

The proposed system consists of sensors and each sensor covers different coverage areas (each sensor has a 30 to 35 m range of coverage). To achieve better connectivity (which means more coverage area i.e., above 1 acre) with no data loss and long-range communication. And also if the length (L) of the sub-field is low, then the received sensor data is more accurate, which means rises monitoring of the agriculture field and reducing the cost-effectiveness of sensor deployment. So total agriculture field is divided into five sub-fields based on wireless connectivity of sensors :

$Total\_agriculture\_field(above\_1\_acre) = 5 \times Subfields$..............................(6)

And each sub-field is having three AWMNs and also each AWMN contains 4 sensors that are deployed in the agriculture field, so the number of sensors selected in a particular agriculture sub-field for coverage connectivity in this equation :

$Sn = \frac{6a}{\pi r2}$........................(7)

where SN=Number of selected sensors, a= Field area, r =Sensor radius for transmission.

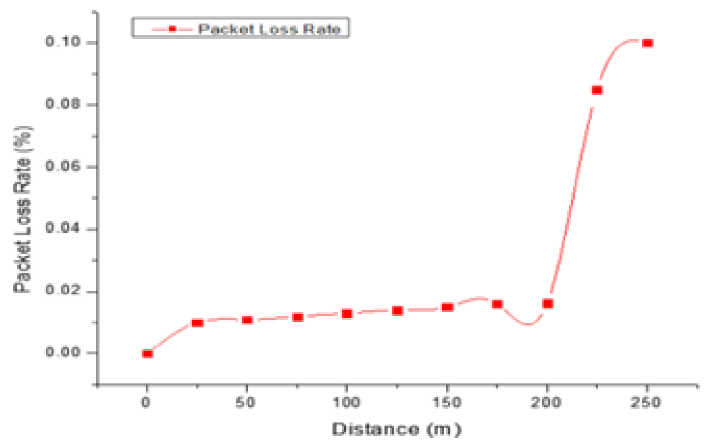


FIG. 3.8 – Packet loss rate versus distance [86]

3.8 Applications of the proposed methodology

Agriculture based on IoT with artificial intelligence is the next emerging thing in smart agriculture. In several real-life applications, IoT is involved. In smart agriculture, using various sensors can monitor agricultural activities like plant and irrigation monitoring. Compared to traditional farming methods, farming based on IoT is much more productive. IoT sensor systems require being simple to use to facilitate farmers to take advantage of them. Some of the applications of the proposed methodology are listed below.[68] :

3.8.1 Plant and crop management

The proposed model with IoT offers a suitable and controllable environment to grow crops [68]

3.8.2 Crop disease and pest management

Combining this model with IoT is used for monitoring, modeling, predicting, and managing disease in farming lands in real-time. This methodology is further used for selecting proper pesticides for protecting crops from infections and hence minimizing the work of farmers [68]. Generally, the reliability of crop disease monitoring and pest management depends on three aspects : sensing, evaluating, Treatment [8].

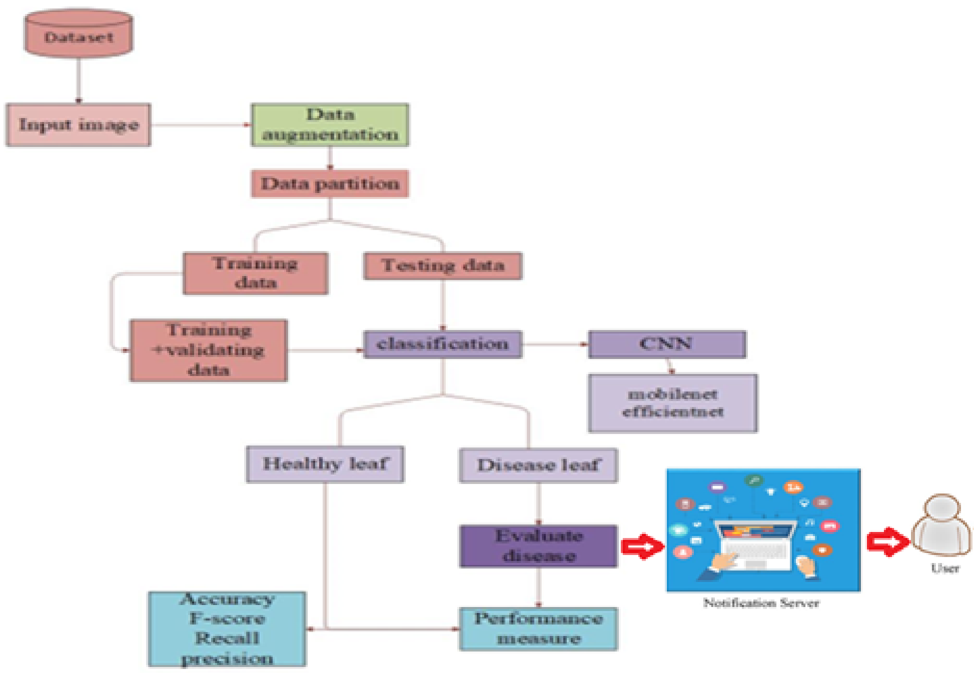


FIG. 3.9 – Diagram of classification plant disease [56].

3.8.3 Crop recommendation[106]

The parameters like Nitrogen, Phosphorous, Potassium, pH, and rainfall are measured using analytical sensors and stored in the database. The compiled data is stored in the cloud. The attained model is ready to recommend the crop for the user for irrigation.

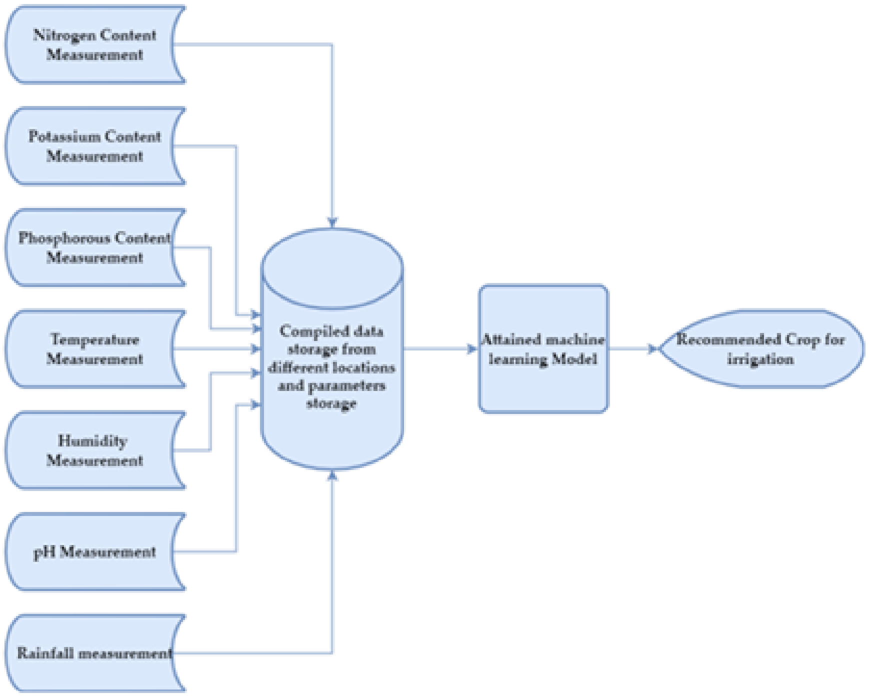


FIG. 3.10 – Diagram of recommended crop for irrigation. [106]

3.8.4 Soil monitoring[8]

The factors that are critical to analyzing the soil nutrient levels include soil type, cropping history, fertilizer application, irrigation level, topography, etc. These factors give insight

regarding the chemical, physical, and biological statuses of soil to identify the limiting factors such that the crops can be dealt with accordingly.

3.8.5 Fertilizer system

It is a natural or chemical substance that can provide important nutrients for the growth and fertility of plants. Plants mainly need three key macronutrients : nitrogen (N) for leaf growth ; phosphorus (P) for root, flowers, and fruit development ; potassium (K) for stem growth and water movement [107].

Any sort of nutrient deficiency or applying them improperly can be seriously harmful to the plant's health. More importantly, excessive use of fertilizer not only results in financial losses but also creates harmful impacts on the soil and environment by depleting the soil quality, poisoning groundwater, and contributing to global climate changes. Overall, crops absorb less than half the nitrogen applied as fertilizer, while the remaining are either emitted to the atmosphere or lost as runoff.

Unbalanced use of fertilizer leads to an imbalance in both soil nutrient levels and global climate as, reportedly, around 80% of the world's deforestation has occurred due to agricultural practices alone [108]. Fertilization helps to precisely estimate the required dose of nutrients.

3.8.6 Irrigation system

Various controlled irrigation methods, like drip irrigation and sprinkler irrigation, are being promoted to tackle water wastage issues, which were also found in traditional methods like flood irrigation and furrow irrigation. Both the crop quality and quantity are badly affected when facing water shortage, as irregular irrigation, even excess, leads to reduced soil nutrients and provokes different microbial infections.

It is not a simple task to accurately estimate the water demand of crops, where factors like crop type, irrigation method, soil type, precipitation, crop needs, and soil moisture retention are involved. Considering this fact, a precise soil and air moisture control system using wireless sensors not only makes optimal use of water but also leads to better crop health [8].

The data collected by the sensor are transmitted to the receiver side circuit. Some threshold value is set at the receiver side microcontroller. If the sensor received is below the threshold value, then the motor (pump) will be in OFF condition and if the received sensor is above the threshold value, then the motor will be turned ON and automatically irrigates the field [109].

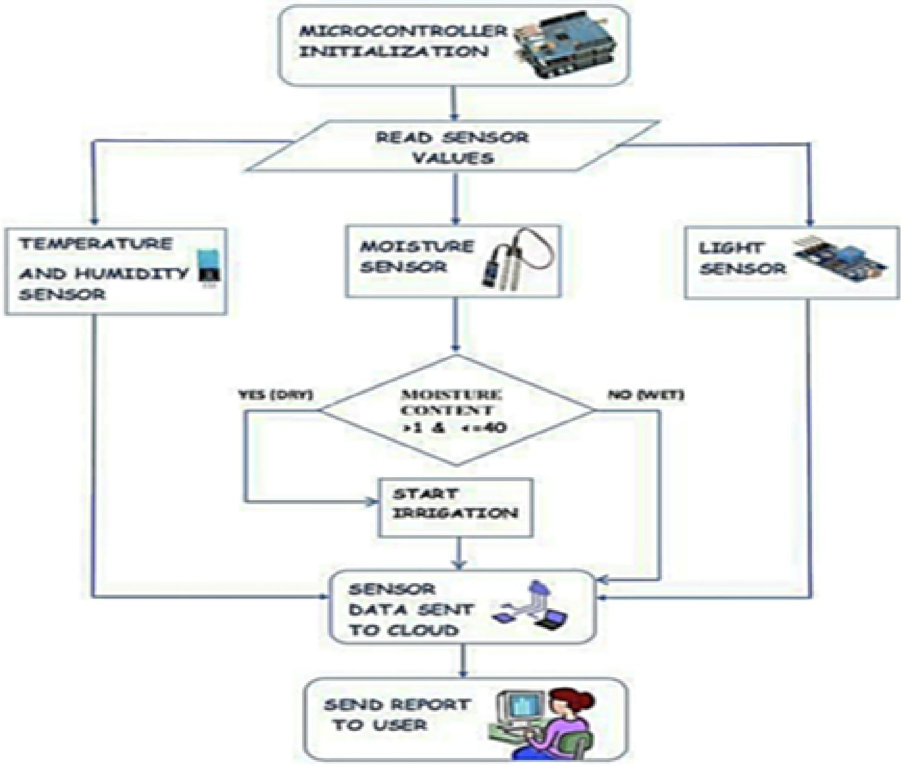


FIG. 3.11 – Schematic diagram of irrigation system. [104]

3.8.7 Weed detection.

3.8.8 Advantages of the system[110]

- The stored data can help other farmers in the future to control the harvest of the crop well in advance of yield.
- Adding more data will make the forecast even more accurate.

3.9 IoT sensors

3.9.1 NPK sensor

It is used to measure the amount of nitrogen,phosphorus, and potassium levels present in the soil, which acts as an alternative to soil moisture sensors.

3.9.2 The DHT

It is a fundamental, ultra simplicity modernized temperature and humidity sensor. It uses a capacitive moisture sensor and discharges a propelled sign on the data stick (no straightforward information pins required). [111].Which generates calibrated digital output. It can be interfaced with any microcontroller like Arduino, Raspberry Pi, etc., and get instantaneous results. It is a low cost which provides high reliability and long-term stability.[112]

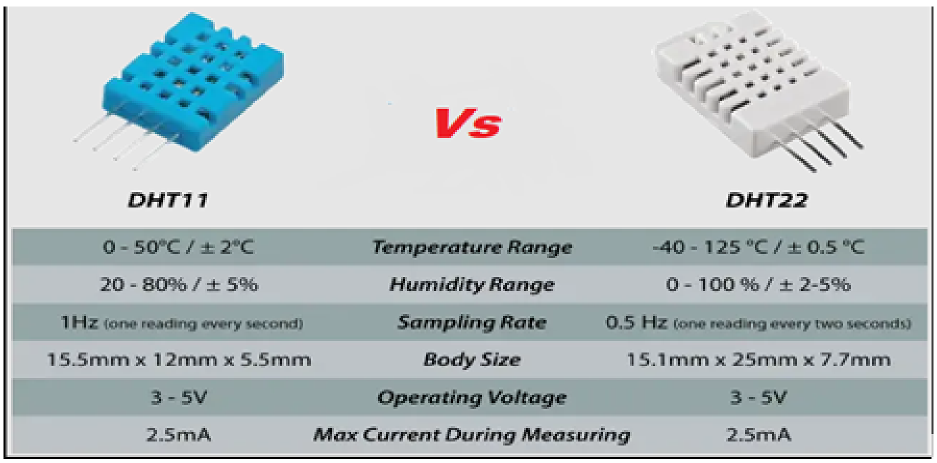


FIG. 3.12 – Comparison between Specifications and features of DHT11 and DHT22(or AM2302) sensor [113].

3.9.3 Soil Moisture Sensor[35]

It is a sort of electrical sensor that can be obtained for a low cost and is used to measure the amount of soil moisture present. This sensor is capable of measuring the soil's total water volume. The two basic components of this sensor are the sensing probes and the sensor module. Sensing probes constitute the initial component. After allowing the current to flow through the soil, the resistance value is measured and compared to the soil's moisture content. The sensor module processes the data read from the sensor probes and then converts the processed data into a digital or analog output. The soil moisture sensor can therefore provide both digital output (D0) and analog output (A0).

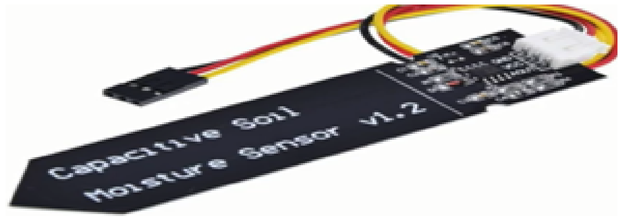


FIG. 3.13 – Soil moisture Sensor.

3.9.4 Rain sensor[114]

The occurrence of rain can be sensed by a rain sensor which consists of a rain detection plate with a comparator that controls intelligence. It also detects short circuits caused by water falling on a printed circuit board ribbon. It works on the principle of resistance. It acts as a variable resistor that changes state. This sensor has two connected outputs which are part of the comparator, one is analog output and the other one is digital output.

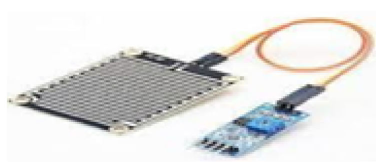


FIG. 3.14 – Rain sensor [114].

3.9.5 Raspberry Pi[115]

It is a credit card-sized minicomputer that can be used with a typical keyboard and mouse and is connected to a computer display or television. With the help of this powerful small gadget, people of all ages may learn about computers and how to develop programs in programming languages like Scratch and Python. It can surf the internet, play games, create spreadsheets and Word documents, and play high-definition video. Additionally, it features all of a desktop computer's features.

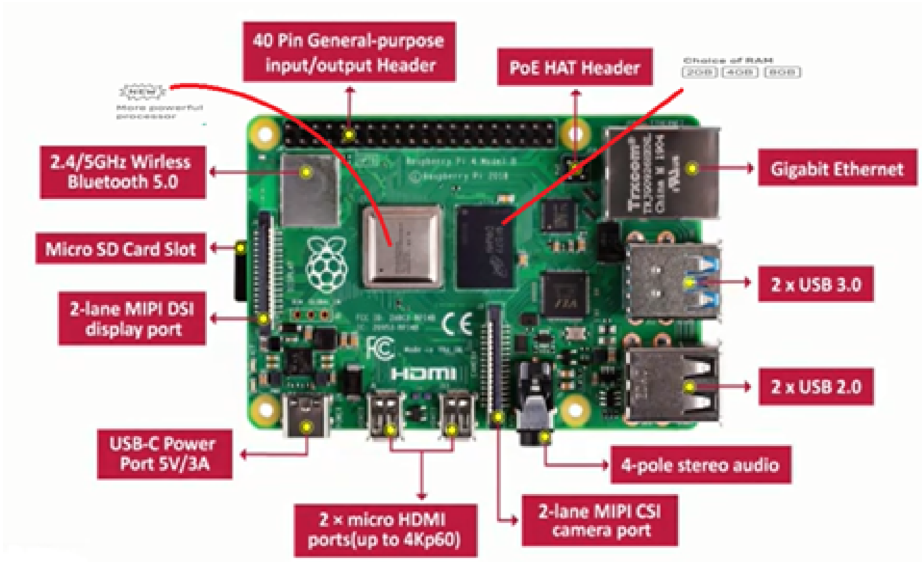


FIG. 3.15 – Raspberry Pi [116].

Arduino Vs Raspberry Pi

This table represent comparison between Arduino and Raspberry Pi :

	Arduino	Raspberry Pi
1	It is development board	It is a mini computer
2	Arduino Uno has ATmega328p controller	ARM controller
3	It is a 8 bit	It is 64bit
4	It has 14 Digital Input/Output Pins	It has 40 GPIO pins
5	It has 6 analog pins	It does not have Analog pins
6	It does not require RTOS	It is based on the Raspbian OS
7	It has on board 5v and 3.3v pins	It also has 5v and 3.3v pins on board
8	It has no on-board Bluetooth module	Pi-4 has on-board Bluetooth
9	It has no on-board Wifi module	Pi-4 has on-board Wifi-module
10	It does not have USB ports on board	Pi-4 has 4 USB ports on-board
11	It does not have Ethernet ports on board	Pi-4 has ethernet ports on-board
12	Arduino can provide onboard storage.	Raspberry Pi don't have storage on board. It provides an SD card port.
13	Arduino require host machine to program	Raspberry pi does not require
14	Arduino uses Arduino, C/C++.	programming language is python but C, C++, Python, ruby are pre-installed.
15	Individual I/O pins in Arduino can drive 40mA	Raspberry Pi GPIO pins can each drive a maximum of 16mA
16	Arduino consumes much less power (~50 mA idle)	Raspberry Pi (700+ mA)
17	It has 2KB SRAM	It has 1GB to 8GB
18	It is clocked at 16MHz	Pi-4 runs at 1.5GHz quad-core
19	Arduino low cost hardware	Pi costs little higher than arduino
20	Does not have camera port	CSI camera port
21	Does not have HDMI port	It has HDMI interface

FIG. 3.16 – Arduino Vs Raspberry Pi . [117]

3.9.6 NODEMCU[35]

It is an open-source IoT platform that is available at a minimal cost. It initially consisted of both software and hardware, the former of which was based on the ESP-12 module and the latter of which ran on Espressif Systems' ESP8266 Wi-Fi System-on-Chip (SoC). Support for the ESP32 32-bit MCU was eventually implemented later on. There are open-source prototyping board designs available for use with the NodeMCU firmware, which is itself an open-source project. Both the firmware and the designs for the prototyping boards are available for free online. It is a Lua-based open-source firmware that was developed specifically for IoT applications. ESP-12E is the module that is based on the ESP8266 MCU and it is the module that is responsible for running this firmware.

It offers Wi-Fi at 2.4 GHz and is compatible with WPA2 and WP2.This table represents the specifications of Node MCU

ESP8266 is used to communicate the studied data from the host system to the cloud platform for further analysis.

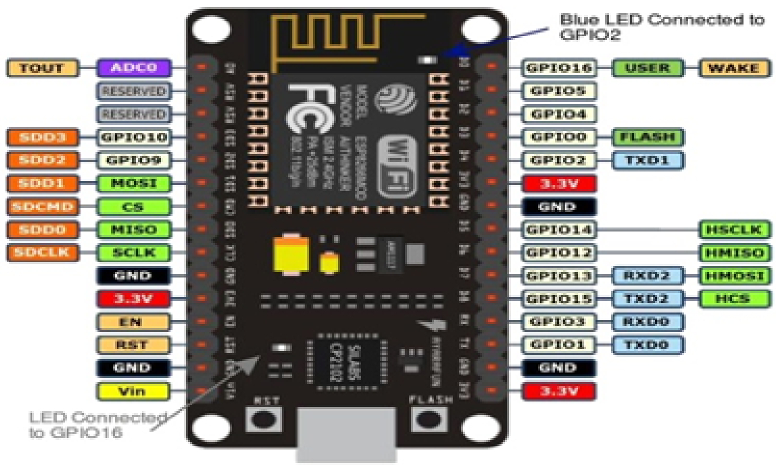


FIG. 3.17 – WiFi module [114].

3.10 Software and Libraries used

3.10.1 Python [110]

It is a high-level, general-purpose programming language. Its design philosophy prioritizes code readability and often employs indentation. Python has dynamic typing and garbage collection. In addition to structured programming (particularly this), it supports a number of other programming paradigms, such as procedural, object-oriented, and functional programming. It is commonly referred to as a "batteries included" language because of its large standard library. Python is commonly used in AI applications, and it benefits from packages like :

Pandas

It is a well-known Python data analysis package. It has nothing to do with ML specifically. As we already know, the data set needs to be ready before training. Given that it was created especially for data preparation and extraction, it is helpful in this situation. High-level data structures, a large range of tools for data analysis, and several built-in techniques for grouping, merging, and filtering data are all provided.

Numpy

It is a well-known Python module for processing big multidimensional arrays and matrices with the aid of a significant number of sophisticated mathematical operations. For ML's basic scientific computations, it is quite helpful. Its skills in linear algebra, the Fourier transform, and random numbers are particularly advantageous. Numpy is used internally by high-end frameworks like TensorFlow to manipulate tensors.

Matplotlib

It is a widely used Python data visualization toolkit. It is not directly connected to ML, like Pandas. It is especially helpful when a coder needs to see the data's patterns. It is a library for 2D charting used to produce 2D graphs and charts. For data visualization, it offers a variety of graphs and plots, including histograms, error charts, bar charts, etc.

Seaborn

It is a library mostly used for statistical plotting in Python. It is built on top of matplotlib.

It provides beautiful default styles and color palettes to make statistical plots more attractive.

Sklearn

It is mainly used for modeling data and it provides efficient tools that are easy to use for any kind of predictive data analysis. The main use cases of this library can be categorized into 6 categories which are the following : pre-processing, regression, classification, clustering, model selection, and dimensionality reduction.

3.10.2 TensorFlow[35]

It is a widely used Python data visualization toolkit. It is not directly connected to ML, like Pandas. It is especially helpful when a coder needs to see the data's patterns. It is a library for 2D charting used to produce 2D graphs and charts. For data visualization, it offers a variety of graphs and plots, including histograms, error charts, bar charts, etc.

• **Open-source :** which means that programmers can simply add more functions and make it more compatible with a variety of datasets.

• **Easy to build models :** It gives you the flexibility to use several levels of abstraction according to your specific requirements. You may utilize the distribution approach on a variety of hardware configurations for large training projects without having to change the model.

• **Powerful experimentation for research :** Users can construct and train sophisticated models with the help of TensorFlow, without having to compromise on speed or performance.

3.10.3 PyTorch

It is a well-known open-source machine-learning library for Python built on Torch, a machine-learning framework with a Lua wrapper that was created in C. It supports several different machine learning (ML) algorithms, such as computer vision, natural language processing (NLP), and many more. It helps in the creation of computational graphs and allows GPU-accelerated Tensor calculations for developers.

3.10.4 Scikit-learn

The majority of supervised and unsupervised learning algorithms are supported, and it is built on top of Numpy and Scipy, two essential Python libraries. It is a great tool for ML beginners because it can also be used for data mining and analysis.

3.10.5 Keras

It comes with several integrated methods for collecting, integrating, and filtering data. This Python machine-learning library is well recognized. Theano, CNTK, or TensorFlow can all utilize this high-level neural network API. The CPU and GPU may both function without any problems. For those new to machine learning, creating a neural network from scratch and designing it is quite helpful. The ease and speed with which Keras facilitates prototyping is one of its best qualities.

3.10.6 OpenCV[79]

It is a free library with support for Windows, Linux, Mac OS, iOS, Android, C++, Python, and Java interfaces that may be used for both academic and commercial purposes. It was created with a significant emphasis on real-time applications and computational efficiency. The library, which may utilize multi-core processing, was written in optimized C/C++ (http ://opencv.org).

3.11 Platforms Used

3.11.1 Google Colab Notebook

It is a collaborative and cloud-based environment for data science and ML projects. It is the main characteristic are :
Jupyter Notebooks, Free GPU and TPU, Integrated Environment(including NumPy, Pandas, Matplotlib, sci-kit-learn, TensorFlow...), Persistent Storage(offers persistent storage in the form of Google Drive), Collaboration(Colab enables collaboration by allowing multiple users to work on the same notebook simultaneously), GitHub Integration.

- RAM : The RAM allocation can range from around 12 GB up to 25 GB or more.
- CPU : they typically include multi-core processors with decent processing power.
- GPU : which can be used to accelerate computations, particularly for DL tasks. The available GPU models can vary, but commonly used GPUs include NVIDIA Tesla K80, Tesla T4, and Tesla P100. These GPUs offer parallel processing capabilities, allowing for faster training and inference of DL models compared to using CPU-only resources.

3.11.2 Kaggle Notebook

It provides an environment for data science and machine learning projects. Here are some of the key characteristics of Kaggle notebooks :
Jupyter Notebooks, Free GPU and TPU, Integrated Environment(including NumPy, Pandas, Matplotlib, sci-kit-learn, TensorFlow...), Persistent Storage(offers persistent storage in the form of Google Drive), Collaboration(Colab enables collaboration by allowing multiple users to work on the same notebook simultaneously), Data access.

The difference between Google Colab vs Kaggle notebooks

It's worth noting that both Google Colab and Kaggle notebooks provide an interactive Jupyter Notebook environment, support for code execution, markdown cells for documentation, and visualization capabilities.

The choice between them depends on your specific needs and preferences. If you primarily work individually and require tight integration with Google services, Google Colab may be a suitable choice. On the other hand, if you prefer collaborative work, access to Kaggle-specific resources, or participation in Kaggle competitions, Kaggle notebooks would be more appropriate.

3.12 Algorithm prediction of plant disease using sensor and camera

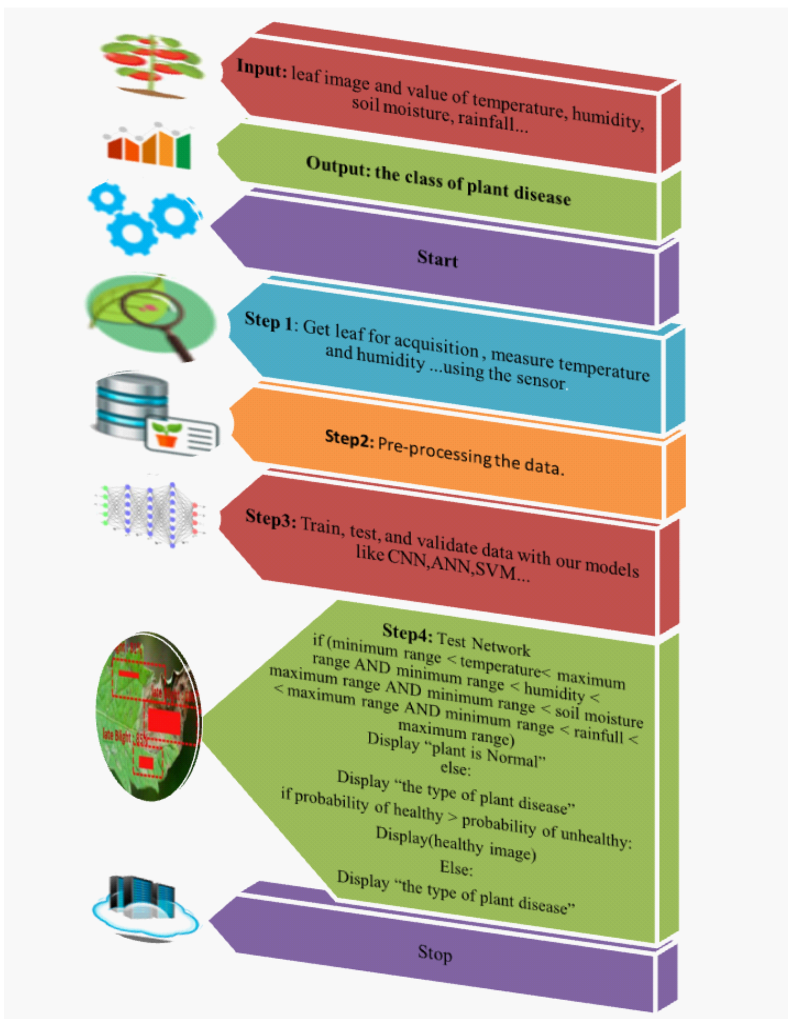


FIG. 3.18 – Algorithm of prediction plant disease.

3.13 Training Models [35]

The camera will be enabled to photograph the plant's leaves. These images will then be submitted to a CNN model, which will use them to predict the type of illness that will develop on the plant. The CNN model is trained in a Kaggle notebook with a lot of images using the TensorFlow framework. It then searches the database and the internet cloud for potential cures for the sickness and delivers the pertinent information to farmers so they can treat the ailment at an earlier stage.

The remedies are delivered to the farmer via the registered mobile number, and the farmer is encouraged to place an order for the required fertilizers. If the model does not predict any disease, an error message is sent to the model, instructing it to revisit the training dataset and update the details based on the new information. By undergoing training with two distinct models, one can achieve a more precise output. By going through these two steps, the system's ability to make accurate judgments is enhanced.

In the first stage, the model is validated using only the sensor values from the sensors. In the second stage, however, the CNN model is trained using images from the camera in addition ANN is trained using the sensor readings, this model is trained in Google Colab.

We train other models for crop recommendation, pest detection, weed detection, and fertilizer system.

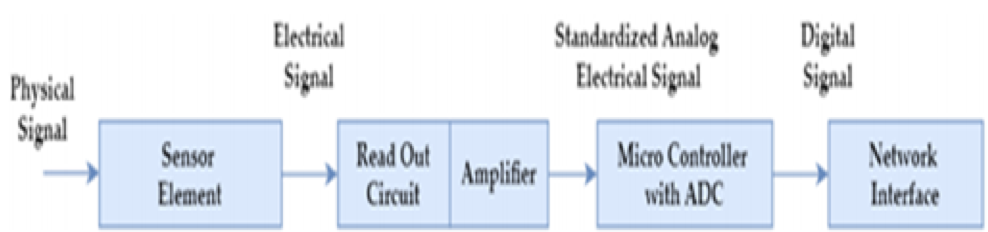


FIG. 3.19 – How information pass in sensor. [106]

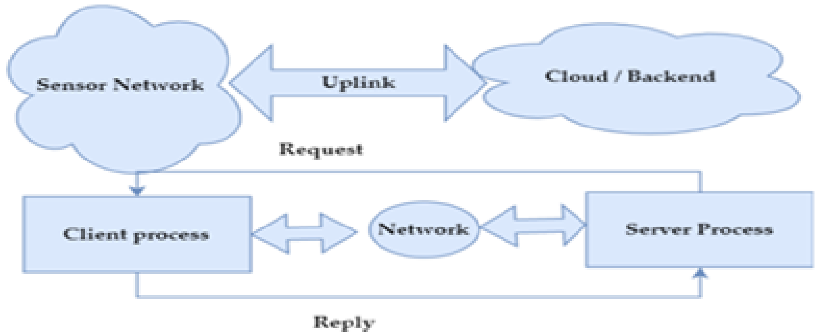


FIG. 3.20 – Proposed IoT Client-server model. [106]

3.14 Dataset Description

There are several databases used in agriculture with the aim of identifying plant diseases, some of which are openly accessible. like as

- The Nutrient Use and Outcome Network (NUOnet)
- The PlantVillage Dataset is a well-known public standard library and widely used database that is available without charge and used for the development and testing of models for the categorization of plant diseases. The database is furthermore often used. There are 54306 pictures of plant leaves in it, and a total of 38 different class designations have been given to them. A crop disease pair appears on each class label.

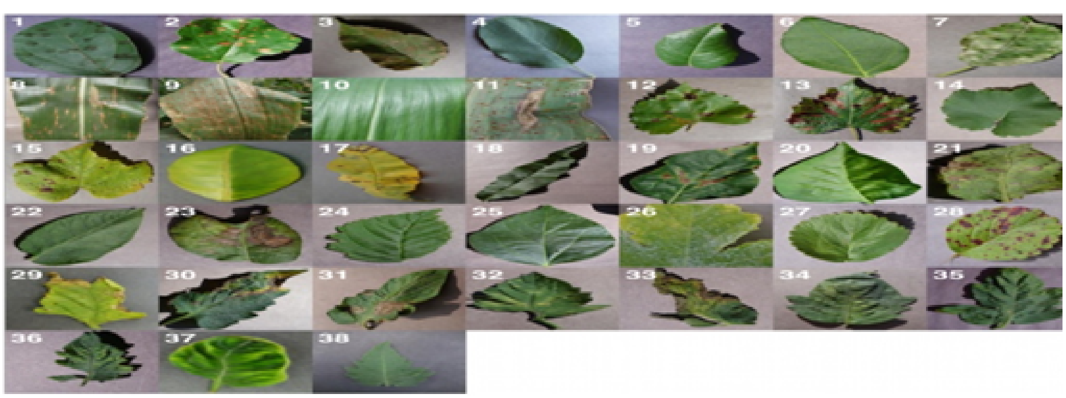


FIG. 3.21 – Example of leaf images from the PlantVillage dataset, representing every crop-disease pair used . [108]

- New Plant Diseases Dataset [118] :

It is an RGB Image dataset containing 3GB of different healthy and unhealthy crop leaves, which is categorized into 38 different classes. The total dataset is divided into an 80/20 ratio of training and validation set preserving the directory structure. A new directory containing 33 test images is created later for prediction purposes.

- Pest Dataset :[119]

a. Type of dataset : Image Dataset

b. No. of training images : 300 images per pest

c. No. of testing images : 50 images per pest

d. Data Source : Automatic script to scrape images of pests from Google through Selenium and Chrome Driver

e. Pests : aphids, armyworm, beetle, bollworm, grasshopper, mites, mosquito, sawfly, stem borer

3.14.1 Dataset preparation

To accomplish the prediction objective of this study, I used PlantVillage Dataset which is partitioned into train, test, and validation subsets using an 80- 10-10 splitting ratio. As a result, there are a total of 16638 images in the training set (i.e., 80% of the available dataset), 2130 images for testing (i.e., 10% of the total dataset), and another 10% for validation.

Normalization was considered by dividing the pixel values by 255. This was done to make the images more acceptable for the beginning values of the models, which was accomplished by dividing the pixel values by 255. The images were resized to a size of 224 * 224 * 3 pixels, and their dimensions were changed to reflect this.

3.14.2 Problematic situations and indicative cases[77]

Images with considerable leaf partial shading were among their contents.

• Photos that include various things in addition to the leaf or leaves, such as fingers, hands in their whole, shoes, pieces of clothes, etc.

• Pictures in which the leaf only takes up a very little, off-center portion of the frame.Based on the excellent performance of the final product.

• The infrequency of sensor data used to forecast plant diseases.

3.15 Analysis and Design

The process of compiling and evaluating data, finding issues, and breaking down a system into its constituent parts is known as system analysis. System analysis is done to investigate a system or its components to pinpoint its goals. The analysis lays forth the system's proper course of action.

System design is the process of designing a new business system's components or modules to satisfy certain needs before implementing it or replacing an old system. The objective of the system is the main consideration in system design [110].

3.15.1 System Architecture

For the goal of ensuring the system satisfies the demands of its users, architecture diagrams may assist system designers and developers in visualizing the high-level, overarching structure of their system or application [110].

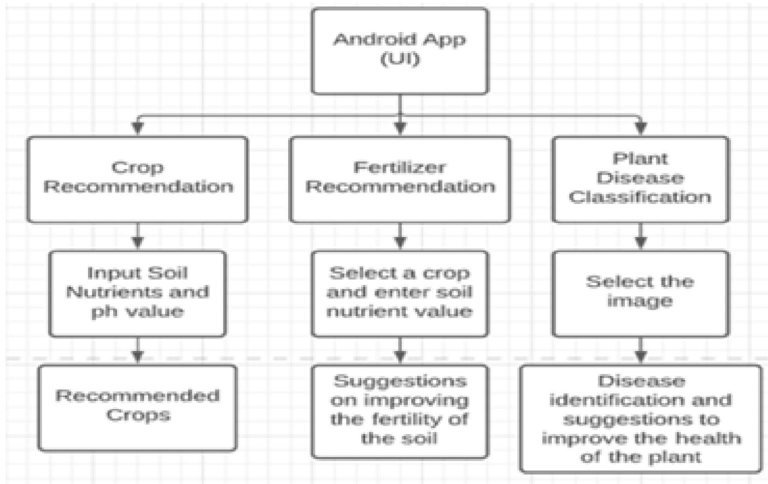


FIG. 3.22 – Architecture of AgriPlantD application. [120]

3.15.2 Use Case Diagram

It is a strategy for locating, outlining, and organizing system requirements in systems analysis. A use case is made up of a collection of potential interactions between people and systems in a particular setting that is connected to a certain objective [110].

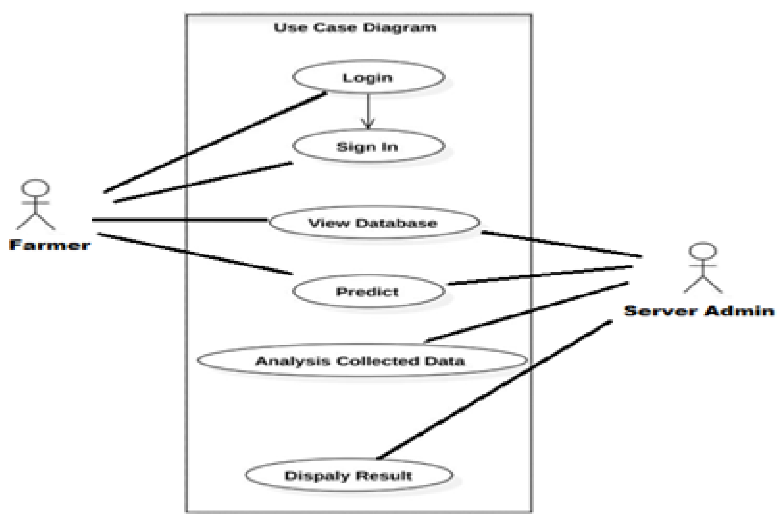


FIG. 3.23 – UML Diagram – I

3.15.3 Sequence Diagram

A sequence diagram simply shows the interaction between objects in sequential order, i.e.the order in which these interactions take place. A sequence diagram simply shows the interaction between objects in sequential order, i.e. the order in which these interactions take place [110].

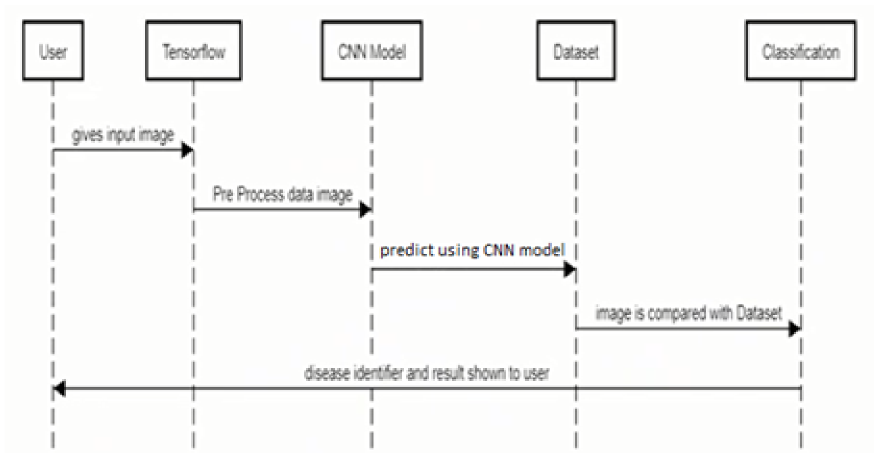


FIG. 3.24 – UML Diagram of predict plant disease by image – II

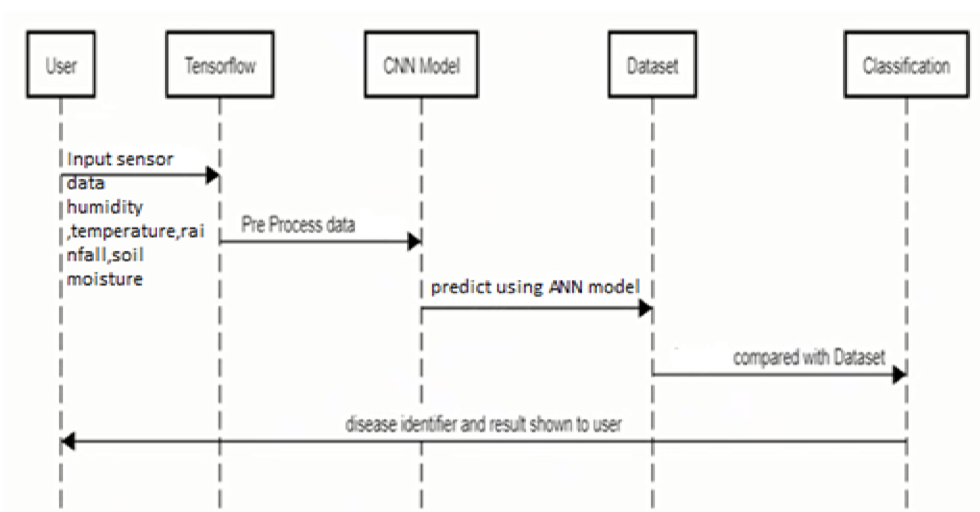


FIG. 3.25 – UML Diagram of predict plant disease by sensor data – III

3.16 Implementation

3.16.1 Results and Discussion

This section gives the experimental and simulation results of the proposed AI-AgriPlantD with IoT system. In addition, it also provides the performance of the proposed CNN model compared to the state-of-the-art approaches using plant leaf datasets. The studies show that plant disease prediction is a domain with promising results. Diverse datasets have been employed, each with its characteristics and associated difficulties : intraclass variability, background diversity, and different lighting and shading conditions during image acquisition [12].

Result of predict plant disease by images

First model EfficientNetB3

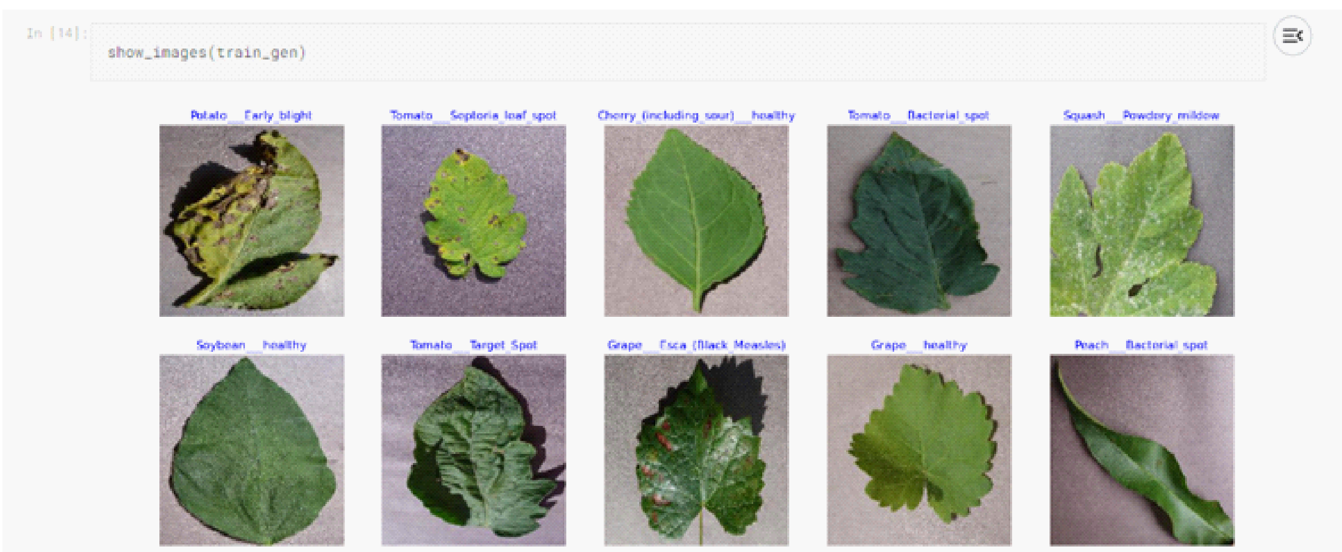


Fig.3.27 Plant Village images.

```
Plantes : ['Tomato', 'Grape', 'Orange', 'Soybean', 'Squash', 'Potato', 'Corn_(maize)', 'Strawberry', 'Peach', 'Apple', 'Bl
ueberry', 'Cherry_(including_sour)', 'Pepper,_bell', 'Raspberry']
----------------------------------------------------------------------------------------------------
Number of plants :  14
diseases : ['Late_blight', 'Haunglongbing_(Citrus_greening)', 'Powdery_mildew', 'Northern_Leaf_Blight', 'Early_blight', 'S
eptoria_leaf_spot', 'Cercospora_leaf_spot Gray_leaf_spot', 'Leaf_scorch', 'Apple_scab', 'Tomato_Yellow_Leaf_Curl_Virus',
'Bacterial_spot', 'Black_rot', 'Powdery_mildew', 'Bacterial_spot', 'Cedar_apple_rust', 'Target_Spot', 'Leaf_blight_(Isario
psis_Leaf_Spot)', 'Late_blight', 'Tomato_mosaic_virus', 'Black_rot', 'Early_blight', 'Common_rust_', 'Esca_(Black_Measle
s)', 'Leaf_Mold', 'Spider_mites Two-spotted_spider_mite', 'Bacterial_spot']
----------------------------------------------------------------------------------------------------
Number of unique diseases :  26
```

FIG. 3.26 – Code source first model.

```
In [15]:
# Create Model Structure
img_size = (224, 224)
channels = 3
img_shape = (img_size[0], img_size[1], channels)
class_count = len(list(train_gen.class_indices.keys())) # to define number of classes in dense layer

# create pre-trained model (you can built on pretrained model such as :  efficientnet, VGG , Resnet )
# we will use efficientnetb3 from EfficientNet family.
base_model = tf.keras.applications.efficientnet.EfficientNetB3(include_top= False, weights= "imagenet", input_shape= img_shap
e, pooling= 'max')

model = Sequential([
    base_model,
    BatchNormalization(axis= -1, momentum= 0.99, epsilon= 0.001),
    Dense(256, kernel_regularizer= regularizers.l2(l= 0.016), activity_regularizer= regularizers.l1(0.006),
                bias_regularizer= regularizers.l1(0.006), activation= 'relu'),
    Dropout(rate= 0.45, seed= 123),
    Dense(class_count, activation= 'softmax')
])

model.compile(Adamax(learning_rate= 0.001), loss= 'categorical_crossentropy', metrics= ['accuracy'])

model.summary()
```

Fig.3.28 Code source first model.

```python
In [ ]:
# Dimension of resized image
DEFAULT_IMAGE_SIZE = tuple((256, 256))

def convert_image_to_array(image_dir):
    try:
        image = cv2.imread(image_dir)
        if image is not None:
            image = cv2.resize(image, DEFAULT_IMAGE_SIZE)
            return img_to_array(image)
        else:
            return np.array([])
    except Exception as e:
        print(f"Error : {e}")
        return None

def predict_disease(image_path):
    image_array = convert_image_to_array(image_path)
    np_image = np.array(image_array, dtype=np.float16) / 225.0
    np_image = np.expand_dims(np_image,0)
    plt.imshow(plt.imread(image_path))
    result = model.predict_classes(np_image)
    print((image_labels.classes_[result][0]))
```

Fig.3.29 Code source first model.

```
Downloading data from https://storage.googleapis.com/keras-applications/efficientnetb3_notop.h5
43941136/43941136 [==============================] - 0s 0us/step
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 efficientnetb3 (Functional)  (None, 1536)             10783535

 batch_normalization (BatchN  (None, 1536)             6144
 ormalization)

 dense (Dense)               (None, 256)               393472

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 38)                9766

=================================================================
Total params: 11,192,917
Trainable params: 11,102,542
Non-trainable params: 90,375
_________________________________________________________________
```

Fig.3.30 Code source first model.

```
In [17]:
history = model.fit(x= train_gen, epochs= epochs, verbose= 0, callbacks= callbacks,
                    validation_data= valid_gen, validation_steps= None, shuffle= False)

Do you want model asks you to halt the training [y/n] ?

 Epoch     Loss   Accuracy  V_loss    V_acc     LR     Next LR  Monitor  % Improv  Duration

2023-06-03 14:51:01.214264: E tensorflow/core/grappler/optimizers/meta_optimizer.cc:954] layout failed: INVALID_ARGUMENT: Size of values 0 does not match size of permutation 4 @ fanin shape insequential/efficientnetb3/block1b_drop/dropout/SelectV2-2-TransposeNHWCToNCHW-LayoutOptimizer

 1 /40     2.995   92.381   0.63604  98.582   0.00100  0.00100  val_loss     0.00    638.16
 2 /40     0.493   98.651   0.34363  99.374   0.00100  0.00100  val_loss    45.97    527.72
 3 /40     0.345   99.148   0.27152  99.484   0.00100  0.00100  val_loss    20.98    540.92
 4 /40     0.278   99.487   0.23495  99.466   0.00100  0.00100  val_loss    13.47    541.19
 5 /40     0.241   99.549   0.20703  99.484   0.00100  0.00100  val_loss    11.88    545.06
enter H to halt training or an integer for number of epochs to run then ask again

Invalid
 6 /40     0.214   99.629   0.18880  99.484   0.00100  0.00100  val_loss     8.80    540.80
enter H to halt training or an integer for number of epochs to run then ask again
```

Fig.3.31 Code source first model.

```
15 /40     0.112    99.979   0.10339  99.779   0.00050  0.00050  val_loss     4.39    548.33
enter H to halt training or an integer for number of epochs to run then ask again

 training will continue until epoch 20
 Epoch     Loss    Accuracy  V_loss    V_acc     LR      Next LR  Monitor  % Improv  Duration
16 /40     0.109    99.961   0.10164  99.871   0.00050  0.00050  val_loss     1.70    544.30
17 /40     0.105    99.956   0.09858  99.797   0.00050  0.00050  val_loss     3.01    544.17
18 /40     0.102    99.965   0.09603  99.742   0.00050  0.00050  val_loss     2.58    544.64
19 /40     0.099    99.965   0.09379  99.797   0.00050  0.00050  val_loss     2.34    533.92
20 /40     0.097    99.961   0.09605  99.724   0.00050  0.00025  val_loss    -2.41    527.56
enter H to halt training or an integer for number of epochs to run then ask again

 training will continue until epoch 22
 Epoch     Loss    Accuracy  V_loss    V_acc     LR      Next LR  Monitor  % Improv  Duration
21 /40     0.092    99.968   0.08644  99.816   0.00025  0.00025  val_loss     7.83    544.55
22 /40     0.089    99.988   0.08652  99.779   0.00025  0.00013  val_loss    -0.08    544.17
enter H to halt training or an integer for number of epochs to run then ask again

training has been halted at epoch 22 due to user input
training elapsed time was 3.0 hours, 28.0 minutes, 11.02 seconds)
```

Fig.3.32 Code source first model.

```
In [18]:
plot_training(history)
```

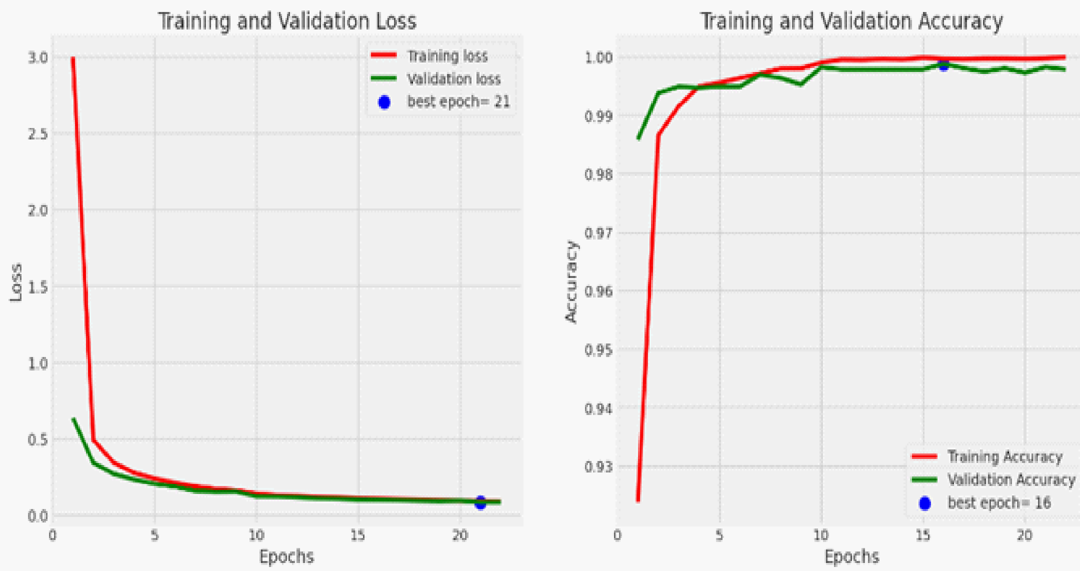


Fig.3.33 Code source first model.

```
print("[INFO] Calculating model accuracy")
scores = model.evaluate(x_test, y_test)
print(f"Test Accuracy: {scores[1]*100}")
```

```
[INFO] Calculating model accuracy
24/24 [==============================] - 1s 30ms/step - loss: 102.7420 - accuracy: 0.7618
Test Accuracy: 76.1842131614685
```

```
In [14]:
# Dump pickle file of the model
print("[INFO] Saving model...")
pickle.dump(model,open('/kaggle/working/plant_disease_classification_model.pkl', 'wb'))
```

```
[INFO] Saving model...
```

```
In [15]:
# Dump pickle file of the labels
print("[INFO] Saving label transform...")
filename = '/kaggle/working/plant_disease_label_transform.pkl'
image_labels = pickle.load(open(filename, 'rb'))
```

```
[INFO] Saving label transform...
```

Fig.3.34 Code source first model.

Evaluate model ¶

```
In [19]:
ts_length = len(test_df)
test_batch_size = test_batch_size = max(sorted([ts_length // n for n in range(1, ts_length + 1) if ts_length%n == 0 and ts_le
ngth/n <= 80]))
test_steps = ts_length // test_batch_size

train_score = model.evaluate(train_gen, steps= test_steps, verbose= 1)
valid_score = model.evaluate(valid_gen, steps= test_steps, verbose= 1)
test_score = model.evaluate(test_gen, steps= test_steps, verbose= 1)

print("Train Loss: ", train_score[0])
print("Train Accuracy: ", train_score[1])
print('-' * 20)
print("Validation Loss: ", valid_score[0])
print("Validation Accuracy: ", valid_score[1])
print('-' * 20)
print("Test Loss: ", test_score[0])
print("Test Accuracy: ", test_score[1])

5431/5431 [==============================] - 161s 30ms/step - loss: 0.0790 - accuracy: 0.9999
5431/5431 [==============================] - 101s 19ms/step - loss: 0.0843 - accuracy: 0.9985
```

Fig.3.35 Code source first model.

```
5431/5431 [==============================] - 161s 30ms/step - loss: 0.0790 - accuracy: 0.9999
5431/5431 [==============================] - 101s 19ms/step - loss: 0.0843 - accuracy: 0.9985
Train Loss:  0.07902366667985916
Train Accuracy:  0.999907910823822
--------------------
Validation Loss:  0.08644238859415054
Validation Accuracy:  0.9981583952903748
--------------------
Test Loss:  0.08428393304347992
Test Accuracy:  0.9985269904136658
```

Fig.3.36 Code source first model.

```
# Classification report
print(classification_report(test_gen.classes, y_pred, target_names= classes))

                                                 precision    recall  f1-score   support

                            Apple___Apple_scab       1.00      0.98      0.99        63
                             Apple___Black_rot       1.00      1.00      1.00        62
                      Apple___Cedar_apple_rust       1.00      1.00      1.00        27
                               Apple___healthy       0.99      1.00      0.99       165
                           Blueberry___healthy       1.00      1.00      1.00       150
      Cherry_(including_sour)___Powdery_mildew       1.00      1.00      1.00       105
             Cherry_(including_sour)___healthy       1.00      1.00      1.00        85
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot   0.94      0.98      0.96        51
                    Corn_(maize)___Common_rust_       1.00      1.00      1.00       119
            Corn_(maize)___Northern_Leaf_Blight       0.99      0.97      0.98        98
                         Corn_(maize)___healthy       1.00      1.00      1.00       116
                              Grape___Black_rot       1.00      1.00      1.00       118
                   Grape___Esca_(Black_Measles)       1.00      1.00      1.00       139
     Grape___Leaf_blight_(Isariopsis_Leaf_Spot)       1.00      1.00      1.00       108
                                Grape___healthy       1.00      1.00      1.00        42
       Orange___Haunglongbing_(Citrus_greening)       1.00      1.00      1.00       551
                         Peach___Bacterial_spot       1.00      1.00      1.00       230
                                Peach___healthy       1.00      1.00      1.00        36
                  Pepper,_bell___Bacterial_spot       1.00      1.00      1.00       100
                         Pepper,_bell___healthy       1.00      1.00      1.00       148
                         Pepper,_bell___healthy       1.00      1.00      1.00       148
                         Potato___Early_blight       1.00      1.00      1.00       100
                          Potato___Late_blight       1.00      1.00      1.00       100
                              Potato___healthy       1.00      1.00      1.00        15
                           Raspberry___healthy       1.00      1.00      1.00        37
                             Soybean___healthy       1.00      1.00      1.00       509
                       Squash___Powdery_mildew       1.00      1.00      1.00       184
                      Strawberry___Leaf_scorch       1.00      1.00      1.00       111
                          Strawberry___healthy       1.00      1.00      1.00        45
                       Tomato___Bacterial_spot       1.00      1.00      1.00       213
                         Tomato___Early_blight       1.00      1.00      1.00       100
                          Tomato___Late_blight       0.99      0.99      0.99       191
                            Tomato___Leaf_Mold       1.00      0.99      0.99        95
                   Tomato___Septoria_leaf_spot       1.00      1.00      1.00       177
 Tomato___Spider_mites Two-spotted_spider_mite       1.00      1.00      1.00       168
                          Tomato___Target_Spot       1.00      1.00      1.00       141
        Tomato___Tomato_Yellow_Leaf_Curl_Virus       1.00      1.00      1.00       536
                  Tomato___Tomato_mosaic_virus       1.00      1.00      1.00        37
                              Tomato___healthy       0.99      1.00      1.00       159

                                      accuracy                           1.00      5431
                                     macro avg       1.00      1.00      1.00      5431
                                  weighted avg       1.00      1.00      1.00      5431
```

Fig.3.37 Code source first model.

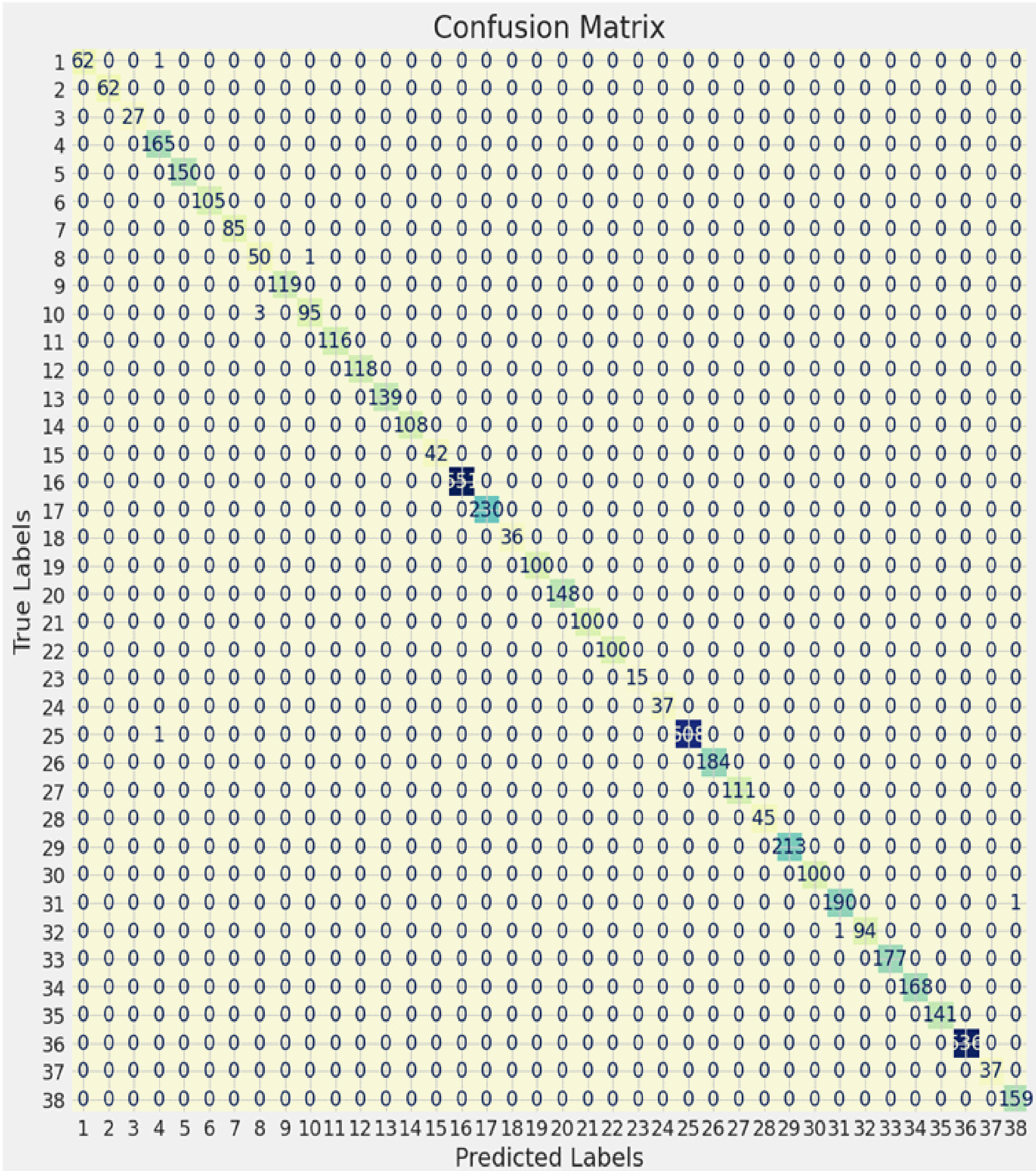


Fig.3.38 Code source first model.

Second model CNN

```python
# Preprocessing data
train_datagen=ImageDataGenerator(rescale=1./255)

val_datagen=ImageDataGenerator(rescale=1./255)

img_width,img_height =256,256
input_shape=(img_width,img_height,3)
batch_size =32

train_generator =train_datagen.flow_from_directory(train_dir,target_size=(img_width,img_height),
                                                   batch_size=batch_size)
val_generator=val_datagen.flow_from_directory(val_dir,target_size=(img_width,img_height),
                                               batch_size=batch_size)
```

```
Found 43444 images belonging to 38 classes.
Found 10861 images belonging to 38 classes.
```

Fig.3.39 Code source second model.

```python
# Model
model = Sequential()

model.add(Conv2D(32, (3, 3), padding="same",input_shape=input_shape))
model.add(Activation("relu"))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(128, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(1024))
model.add(Activation("relu"))
model.add(BatchNormalization())
model.add(Dense(n_classes))
model.add(Activation("softmax"))
model.summary()
```

Fig.3.40 Code source second model.

```python
#### %%time
#loss=MeanSquaredError(),
from tensorflow.keras.metrics import AUC, CategoricalAccuracy, Precision, Recall,MeanSquaredError
from sklearn.metrics import f1_score
metrics = ['accuracy',
           MeanSquaredError(),
    CategoricalAccuracy(),
    Precision(),
    Recall(),
#     AUC()
]
# Model building
opt=keras.optimizers.SGD(lr=0.01)
model.compile(optimizer=opt,loss='categorical_crossentropy', metrics=metrics)
train=model.fit_generator(train_generator,
                          epochs=35,
                          steps_per_epoch=train_generator.samples // batch_size,
                          validation_data= val_generator,
                          validation_steps= val_generator.samples// batch_size,
                           verbose=1)
```

Fig.3.41 Code source second model.

```
precision: 1.0000 - recall: 1.0000 - val_loss: 0.1218 - val_accuracy: 0.9639 - val_mean_squared_error: 0.0015 - val_categorical_accur
- precision: 1.0000 - recall: 1.0000 - val_loss: 0.1211 - val_accuracy: 0.9636 - val_mean_squared_error: 0.0015 - val_categorical_accuracy: 0.9636 - val_precisi
precision: 1.0000 - recall: 1.0000 - val_loss: 0.1210 - val_accuracy: 0.9633 - val_mean_squared_error: 0.0014 - val_categorical_accuracy: 0.9633 - val_precisio
- precision: 1.0000 - recall: 1.0000 - val_loss: 0.1202 - val_accuracy: 0.9639 - val_mean_squared_error: 0.0014 - val_categorical_accuracy: 0.9639 - val_precisi
- precision: 1.0000 - recall: 1.0000 - val_loss: 0.1199 - val_accuracy: 0.9639 - val_mean_squared_error: 0.0014 - val_categorical_accuracy: 0.9639 - val_precisi
- precision: 1.0000 - recall: 1.0000 - val_loss: 0.1210 - val_accuracy: 0.9641 - val_mean_squared_error: 0.0014 - val_categorical_accuracy: 0.9641 - val_precisi
- precision: 1.0000 - recall: 1.0000 - val_loss: 0.1221 - val_accuracy: 0.9632 - val_mean_squared_error: 0.0015 - val_categorical_accuracy: 0.9632 - val_precisi
precision: 1.0000 - recall: 1.0000 - val_loss: 0.1221 - val_accuracy: 0.9626 - val_mean_squared_error: 0.0015 - val_categorical_accuracy: 0.9626 - val_precisio
- precision: 1.0000 - recall: 1.0000 - val_loss: 0.1204 - val_accuracy: 0.9629 - val_mean_squared_error: 0.0014 - val_categorical_accuracy: 0.9629 - val_precisi
```

Fig.3.42 Code source second model.

```
In [26]:
predictions = list(itertools.chain.from_iterable(predictions))
labels = list(itertools.chain.from_iterable(labels))

In [27]:
print("Train Accuracy  : {:.2f} %".format(history.history['accuracy'][-1]*100))
print("Test Accuracy   : {:.2f} %".format(accuracy_score(labels, predictions) * 100))
print("Precision Score : {:.2f} %".format(precision_score(labels, predictions, average='micro') * 10
0))
print("Recall Score    : {:.2f} %".format(recall_score(labels, predictions, average='micro') * 100))

Train Accuracy  : 99.75 %
Test Accuracy   : 98.34 %
Precision Score : 98.34 %
Recall Score    : 98.34 %
```

Fig.3.43Code source second model.

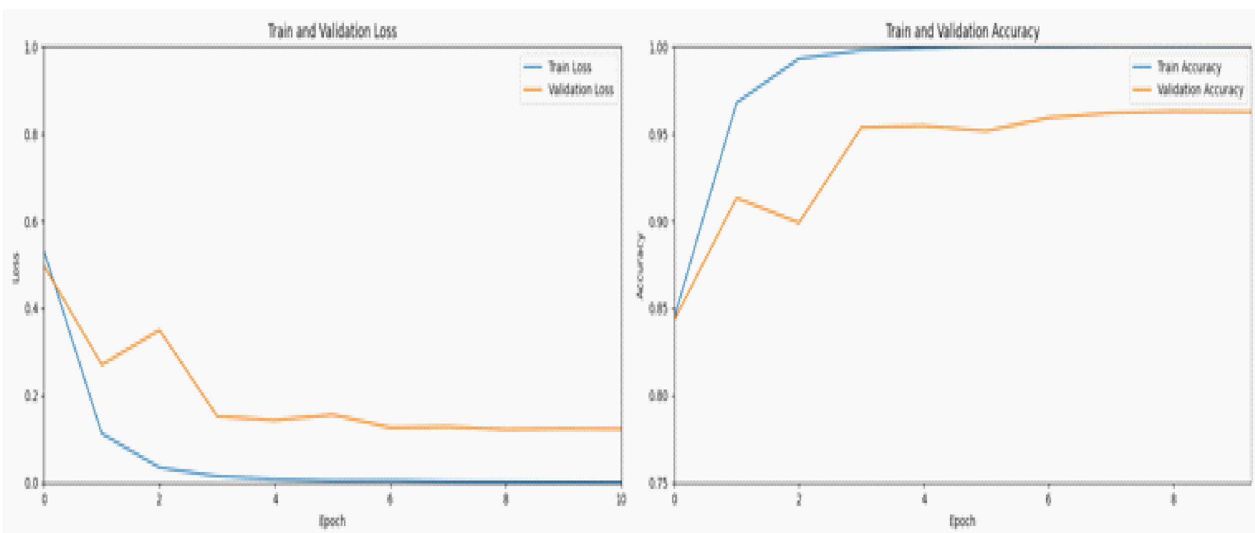


Fig.3.44 Code source second model.

Third model CNN

predict_plant_disease_new

Notebook Input Output Logs Comments (0) Settings

Using ImageDataGenerator to augment data by performing various operations on the training images.

In [7]:
```
augment = ImageDataGenerator(rotation_range=25, width_shift_range=0.1,
                             height_shift_range=0.1, shear_range=0.2,
                             zoom_range=0.2, horizontal_flip=True,
                             fill_mode="nearest")
```

In [8]:
```
print("[INFO] Splitting data to train and test...")
x_train, x_test, y_train, y_test = train_test_split(np_image_list, image_labels, test_size=0.2, rand
om_state = 42)
```

```
[INFO] Splitting data to train and test...
```

Fig.3.45 Code source Third model.

10]:
```
model = Sequential()
inputShape = (HEIGHT, WIDTH, DEPTH)
chanDim = -1

if K.image_data_format() == "channels_first":
    inputShape = (DEPTH, HEIGHT, WIDTH)
    chanDim = 1

model.add(Conv2D(32, (3, 3), padding="same",input_shape=inputShape))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPooling2D(pool_size=(3, 3)))
model.add(Dropout(0.25))
model.add(Conv2D(64, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(64, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Conv2D(128, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(128, (3, 3), padding="same"))
```

Fig.3.46 Code source Third model.

```
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(128, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(1024))
model.add(Activation("relu"))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Dense(n_classes))
model.add(Activation("softmax"))

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 256, 256, 32)      896

 activation (Activation)     (None, 256, 256, 32)      0

 batch_normalization (BatchN  (None, 256, 256, 32)     128
```

Fig.3.47 Code source Third model.

Train Model

In [11]:

```
# Initialize optimizer
opt = Adam(lr=LR, decay=LR / EPOCHS)

# Compile model
model.compile(loss="categorical_crossentropy", optimizer=opt, metrics=["accuracy"])

# Train model
print("[INFO] Training network...")
history = model.fit_generator(augment.flow(x_train, y_train, batch_size=BATCH_SIZE),
                              validation_data=(x_test, y_test),
                              steps_per_epoch=len(x_train) // BATCH_SIZE,
                              epochs=EPOCHS,
                              verbose=1)
```

```
[INFO] Training network...
```

Fig.3.48 Code source Third model.

```
Epoch 143/150
95/95 [==============================] - 41s 433ms/step - loss: 0.4215 - accuracy: 0.8931 - val_
loss: 166.1724 - val_accuracy: 0.6224
Epoch 144/150
95/95 [==============================] - 41s 427ms/step - loss: 0.4442 - accuracy: 0.8954 - val_
loss: 253.7350 - val_accuracy: 0.7303
Epoch 145/150
95/95 [==============================] - 41s 434ms/step - loss: 0.4553 - accuracy: 0.8875 - val_
loss: 1591.6422 - val_accuracy: 0.8329
Epoch 146/150
95/95 [==============================] - 41s 427ms/step - loss: 0.4391 - accuracy: 0.8918 - val_
loss: 118.1974 - val_accuracy: 0.6434
Epoch 147/150
95/95 [==============================] - 41s 430ms/step - loss: 0.4075 - accuracy: 0.9003 - val_
loss: 295.0479 - val_accuracy: 0.7316
Epoch 148/150
95/95 [==============================] - 41s 432ms/step - loss: 0.4427 - accuracy: 0.8954 - val_
loss: 126.9389 - val_accuracy: 0.7395
Epoch 149/150
95/95 [==============================] - 40s 425ms/step - loss: 0.4002 - accuracy: 0.9049 - val_
loss: 136.2873 - val_accuracy: 0.7250
Epoch 150/150
95/95 [==============================] - 41s 433ms/step - loss: 0.4174 - accuracy: 0.8980 - val_
loss: 102.7420 - val_accuracy: 0.7618
```

Fig.3.49 Code source Third model.

```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(acc) + 1)

# Train and validation accuracy
plt.plot(epochs, acc, 'b', label='Training accurarcy')
plt.plot(epochs, val_acc, 'r', label='Validation accurarcy')
plt.title('Training and Validation accurarcy')
plt.legend()

plt.figure()

# Train and validation loss
plt.plot(epochs, loss, 'b', label='Training loss')
plt.plot(epochs, val_loss, 'r', label='Validation loss')
plt.title('Training and Validation loss')
plt.legend()
plt.show()
```

Fig.3.50 Code source Third model.

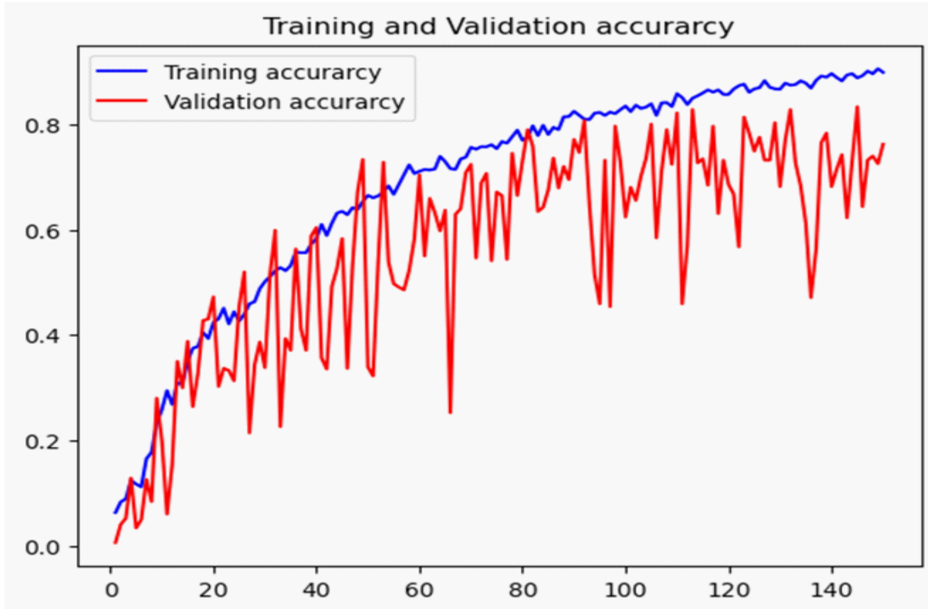


Fig.3.51 Code source Third model.

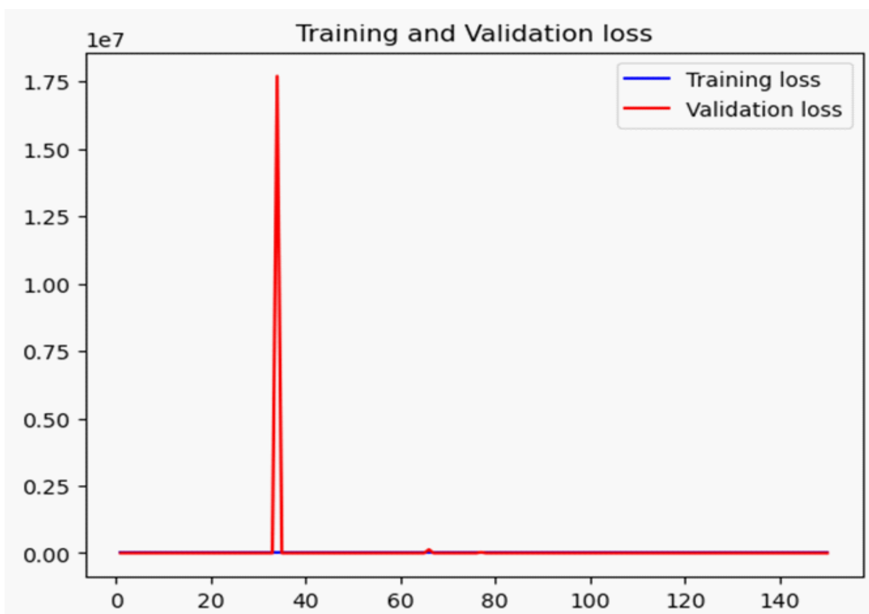


Fig.3.52 Code source Third model.

```
In [13]:
print("[INFO] Calculating model accuracy")
scores = model.evaluate(x_test, y_test)
print(f"Test Accuracy: {scores[1]*100}")
```

```
[INFO] Calculating model accuracy
24/24 [==============================] - 1s 30ms/step - loss: 102.7420 - accuracy: 0.7618
Test Accuracy: 76.1842131614685
```

```
In [14]:
# Dump pickle file of the model
print("[INFO] Saving model...")
pickle.dump(model,open('/kaggle/working/plant_disease_classification_model.pkl', 'wb'))
```

```
[INFO] Saving model...
```

Fig.3.53 Code source Third model.

```
[42]
from tensorflow.keras.utils import img_to_array
# Dimension of resized image
DEFAULT_IMAGE_SIZE = tuple((256, 256))

def convert_image_to_array(image_dir):
    try:
        image = cv2.imread(image_dir)
        if image is not None:
            image = cv2.resize(image, DEFAULT_IMAGE_SIZE)
            return img_to_array(image)
        else:
            return np.array([])
    except Exception as e:
        print(f"Error : {e}")
        return None

def predict_disease(image_path):
    image_array = convert_image_to_array(image_path)
    np_image = np.array(image_array, dtype=np.float16) / 225.0
    np_image = np.expand_dims(np_image,0)
    plt.imshow(plt.imread(image_path))
    result = model.predict(np_image)#_classes
    probabilty = result.flatten()
    max_prob = probabilty.max()
    result = model.predict(np_image)#_classes
    probabilty = result.flatten()
    max_prob = probabilty.max()
    print()
    result=np.argmax(result,axis=1)
    print('Predicted :'+(image_labels.classes_[result[0]])+"  "+str(max_prob*100)[0:4]+" %")
    print('Real :'+ file_name)
```

Fig.3.54 Code source Third model.

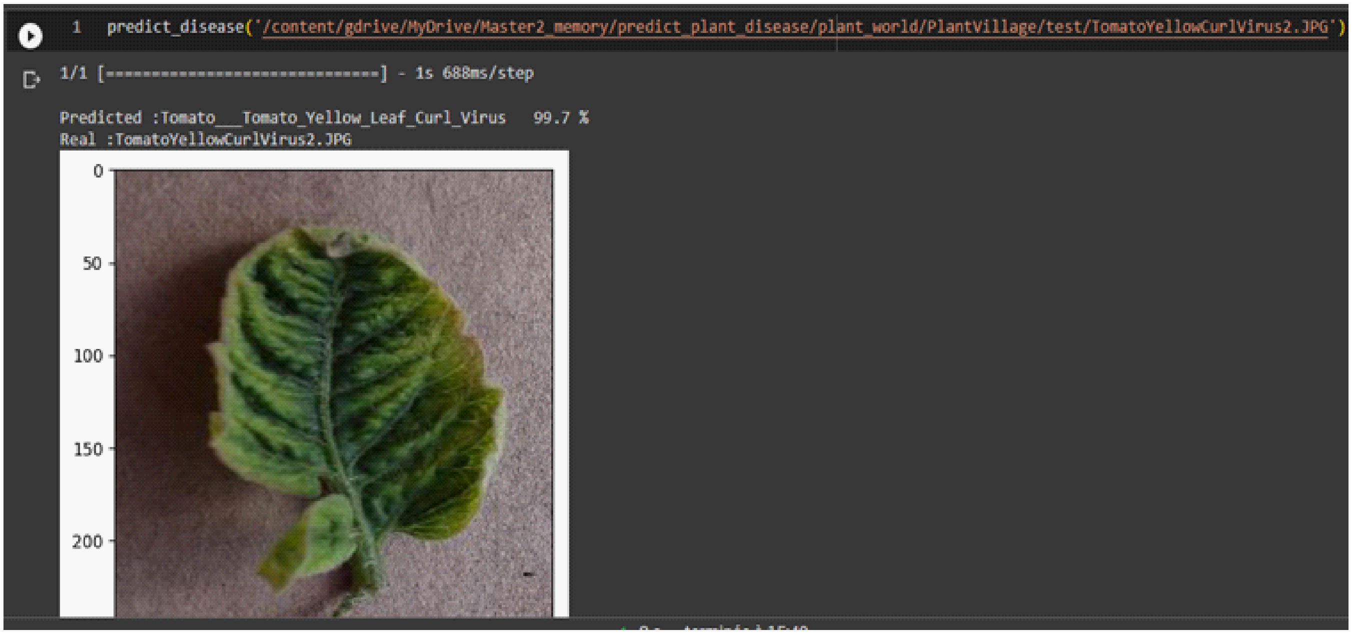


Fig.3.55 Code source Third model.

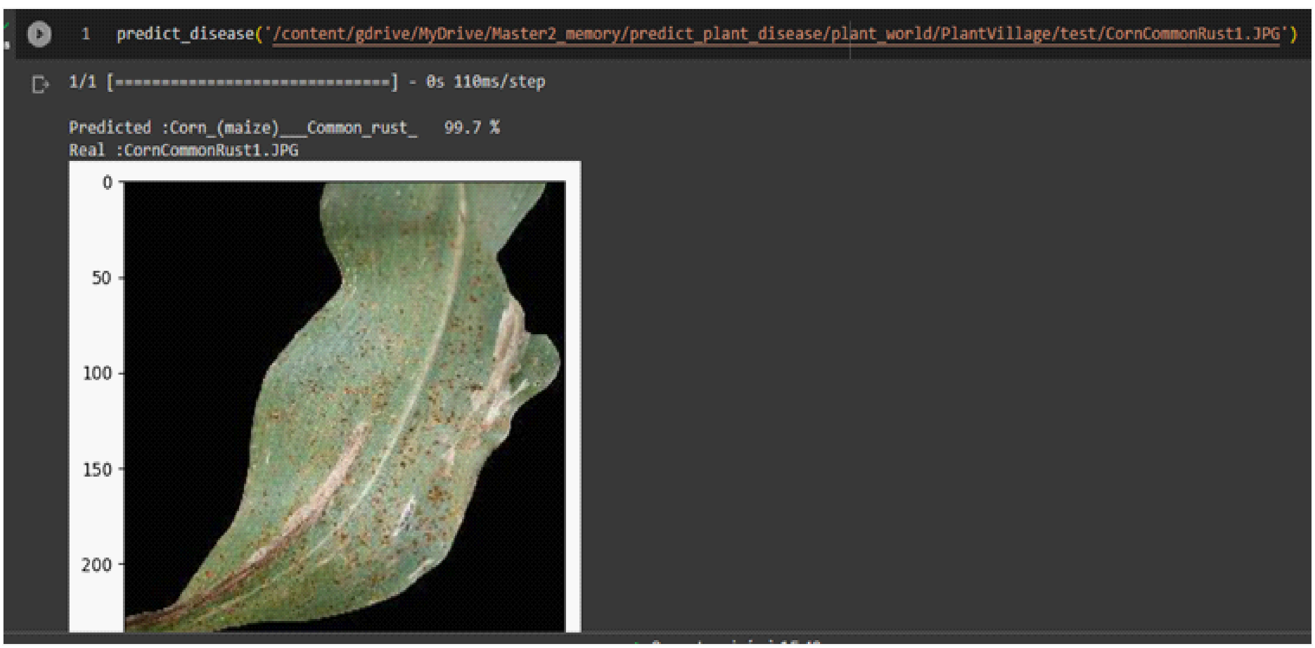


Fig.3.56 Code source Third model.

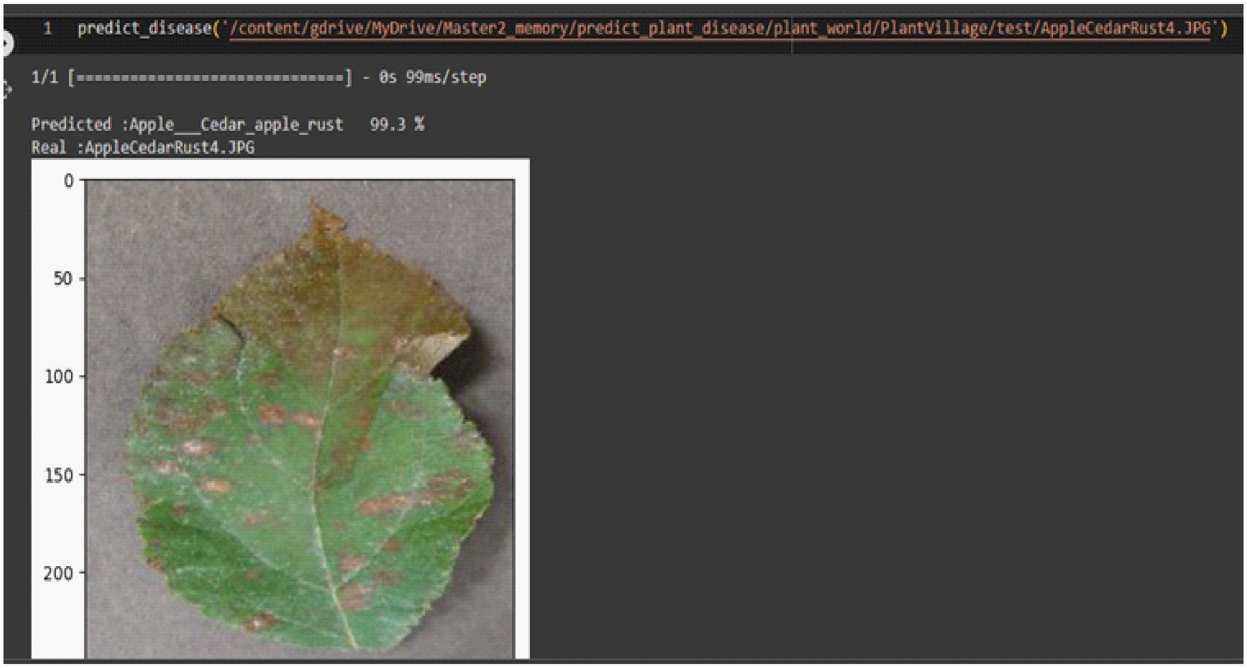


Fig.3.57 Code source Third model.

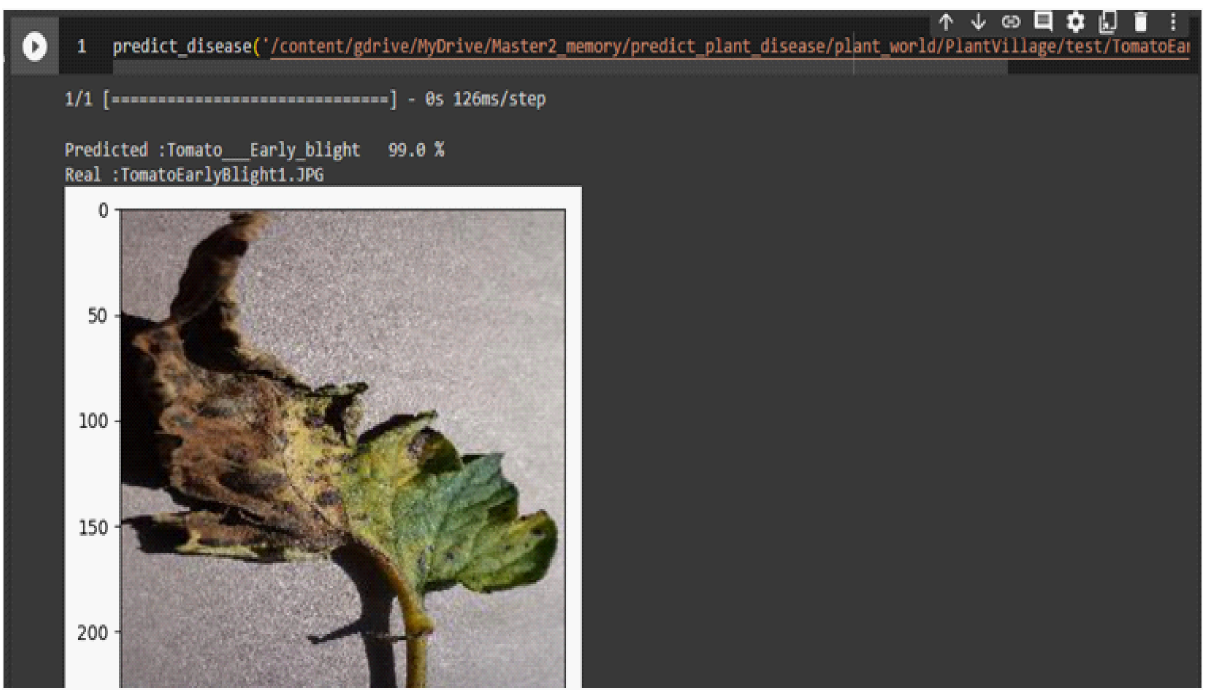


Fig.3.58 Code source Third model.

Discusion

Authors	Model/Algorithm	Dataset	Result
Ferentinos et al., (2018)	CNN	PlantVillage	99.53% accuracy
Reddy et al., (2023)	CNN	PlantVillage	99.2% accuracy
Sladojevic et al. (2016)	DCNN	PlantVillage	Between 91%to98% accuracy
Mukti and Biswas (2019)	ResNet50	PlantVillage	99.80%accuracy
Amara et al. (2017)	CNN (LeNet architecture)	CNN (LeNet architecture)	92-99% accuracy
Mohanty et al. (2016)	CNNs (AlexNet, GoogLeNet)	PlantVillage	99,35% accuracy
Brahimi et al. (2017)	CNN (AlexNet, GoogLeNet)	PlantVillage	99% accuracy
My model :	EfficientNetB3	PlantVillage	99.99% accuracy
My model	CNN	PlantVillage	99.75% accuracy
My model	CNN	PlantVillage	89.80% accuracy

FIG. 3.27 – Comarative study between my models and previous models

As we can see from the table, I got very satisfactory results from the previous studies, as I trained 3 models in different ways, and I aim to obtain greater accuracy when predicting plant disease.

Predict plant disease by climate change

```
#fill null value by value in this i used 0.0 or column means or median
df=df.fillna(0.0)
```

```
df.head()
```

	rainfall	temp	humidity	moisture	disease
0	0.0	10	90	10	Powdery_mildew
1	0.1	11	90	11	Powdery_mildew
2	0.2	12	90	12	Powdery_mildew
3	0.3	13	90	13	Powdery_mildew
4	0.4	14	90	14	Powdery_mildew

Fig.3.60 Code source first model.

```
#missing value tabular
df.isnull().sum()
```

```
rainfall    0
temp        0
humidity    0
moisture    0
disease     0
dtype: int64
```

```
df.isnull().any()
```

```
rainfall    False
temp        False
humidity    False
moisture    False
disease     False
dtype: bool
```

Fig.3.61 Code source first model.

```
#replace null vaalue by value
df.interpolate()
```

	rainfall	temp	humidity	moisture	disease
0	0.0	10	90	10	Powdery_mildew
1	0.1	11	90	11	Powdery_mildew
2	0.2	12	90	12	Powdery_mildew
3	0.3	13	90	13	Powdery_mildew
4	0.4	14	90	14	Powdery_mildew
...	...	...	...	...	...
4505	1.8	46	100	10	Root_rotand_leaf_blight
4506	1.9	47	100	11	Root_rotand_leaf_blight
4507	2.0	48	100	12	Root_rotand_leaf_blight
4508	0.5	49	100	13	Root_rotand_leaf_blight
4509	0.6	50	100	14	Root_rotand_leaf_blight

4510 rows × 5 columns

Fig.3.62 Code source first model.

```
[ ]  print("diseases    :   \n",df['disease'].unique())

diseases    :
 ['Powdery_mildew' 'Anthracnose' 'Rust' 'Root_rotand_leaf_blight']

[ ]  print("number of data for each disease :    \n ",df['disease'].value_counts())

number of data for each disease :
  Root_rotand_leaf_blight    1890
Powdery_mildew             1210
Rust                        780
Anthracnose                 630
Name: disease, dtype: int64
```

Fig.3.63 Code source first model.

```
[ ]  x_train,x_test,y_train,y_test = model_selection.train_test_split(x,Y, test_size=0.2,stratify=Y, random_state=42)
     print("len x train: ",len(x_train),"len y train: ",len(y_train),"len x test: ",len(x_test),"len y test: ",len(y_test))

len x train:  3608 len y train:  3608 len x test:  902 len y test:  902

[ ]  #normalisation for tabular data value between [0,1]

     from sklearn.preprocessing import MinMaxScaler
     Scaler=MinMaxScaler()
     Scaler.fit_transform(x_train)
     x_train.head()
```

	rainfall	temp	humidity	moisture
1695	0.5	15	96	10
754	0.4	16	96	14
3926	1.5	34	94	11
4220	0.5	34	97	10
4213	1.4	48	97	13

Fig.3.64 Code source first model.

```
acc.append(ac)
model.append("Gradient Boosting Classifier")
confusion_gbc=confusion_matrix(y_test,y_pred_gbc)
plt.figure(figsize=(8,8))
sns.heatmap(confusion_gbc,annot=True)
plt.xlabel("Predicted")
plt.ylabel("Actual")
print(classification_report(y_test,y_pred_gbc))
```

```
score : 92.6829268292683  %
                          precision    recall  f1-score   support

          Anthracnose       1.00      0.48      0.65       126
       Powdery_mildew       0.79      1.00      0.88       242
Root_rotand_leaf_blight       1.00      1.00      1.00       378
                 Rust       1.00      1.00      1.00       156

             accuracy                           0.93       902
            macro avg       0.95      0.87      0.88       902
         weighted avg       0.94      0.93      0.92       902
```

Fig.3.65 Code source first model.

```
pd.DataFrame(acc,model)
```

	0
KNeighbors Classifier	0.830377
Support Vector Classifier	0.926829
Logistic Regression	0.895787
Random Forest Classifier	0.862528
Gradient Boosting Classifier	0.926829
LGBM Classifier	0.872506
Decision TreeClassifier	0.926829
Guassian Naive Bayes	0.899113
Support Vector Machine (SVM)	0.926829
Extra Tree Classifier	0.868071
AdaBoost	0.926829

Fig.3.66 Code source first model.

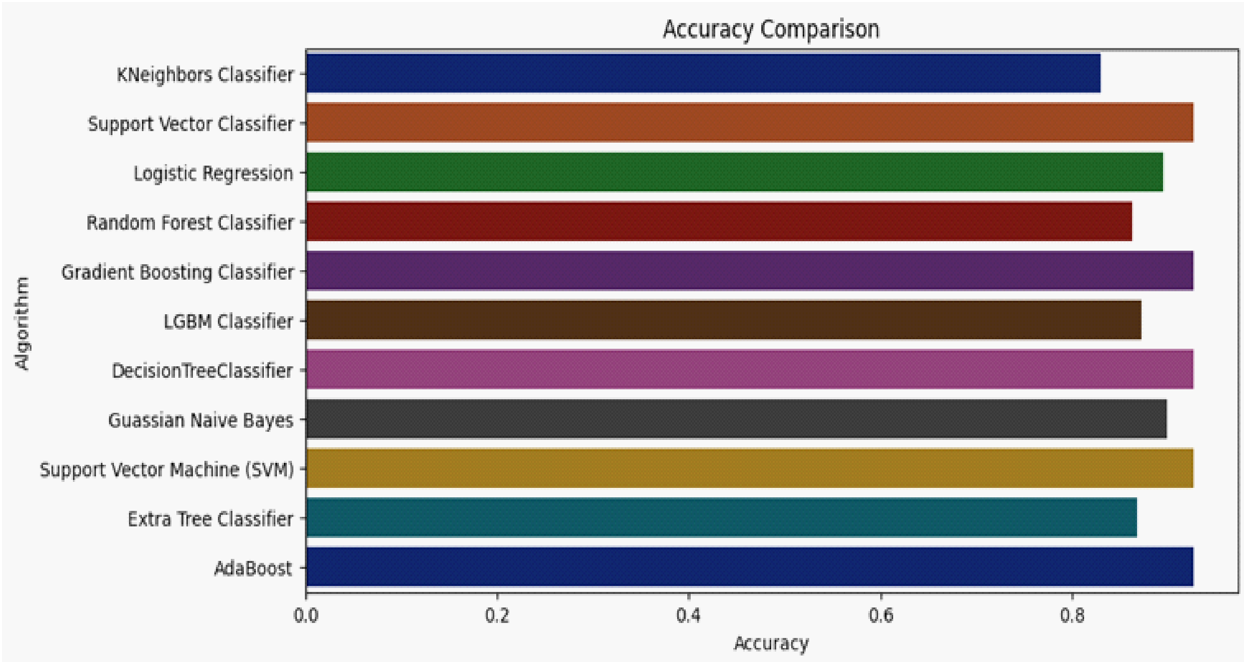
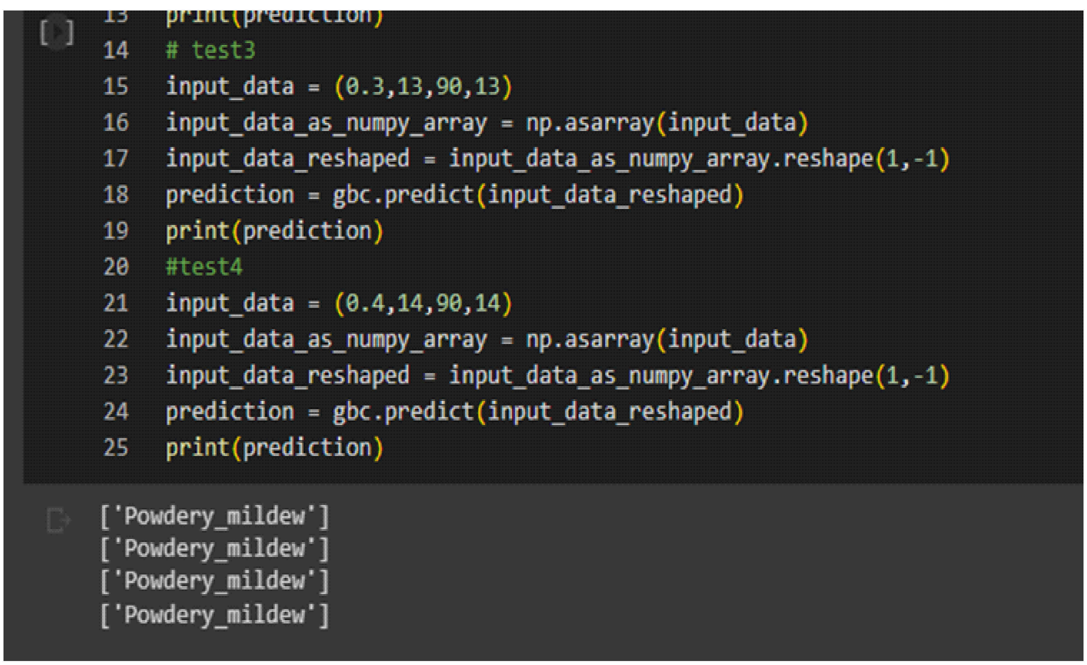


Fig.3.67 Code source first model.

The best result is obtained by Naive Bayes, support vector classifier, Gradient boosting classifier, Decision tree, Support vector machine, and AdaBoost.



```
print(prediction)
# test3
input_data = (0.3,13,90,13)
input_data_as_numpy_array = np.asarray(input_data)
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
prediction = gbc.predict(input_data_reshaped)
print(prediction)
#test4
input_data = (0.4,14,90,14)
input_data_as_numpy_array = np.asarray(input_data)
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
prediction = gbc.predict(input_data_reshaped)
print(prediction)

['Powdery_mildew']
['Powdery_mildew']
['Powdery_mildew']
['Powdery_mildew']
```

Fig.3.68 Code source first model.

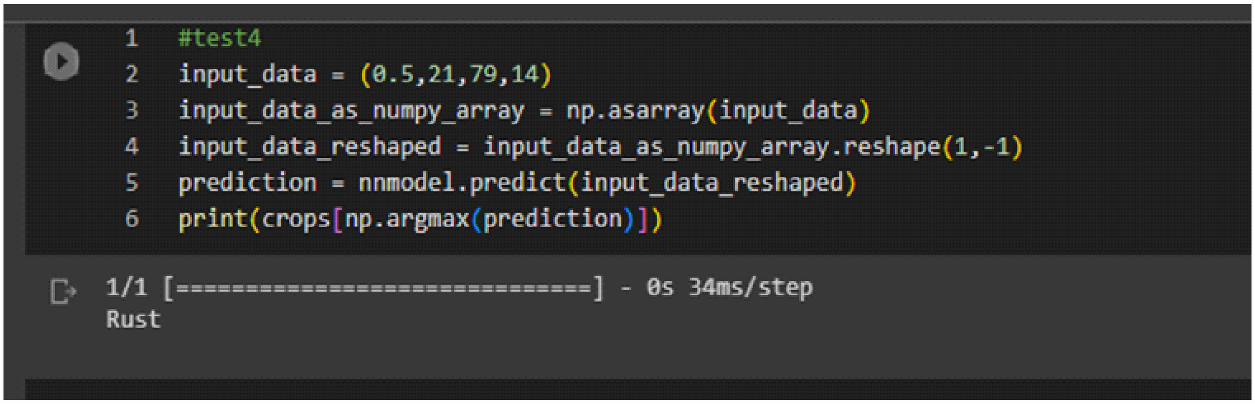
```
[ ]
import tensorflow as tf
nnmodel = tf.keras.models.Sequential([
    tf.keras.layers.Dense(32,activation = 'relu',input_dim = 4),
    tf.keras.layers.BatchNormalization(synchronized=True),
    tf.keras.layers.Dense(64,activation = 'relu'),
    tf.keras.layers.BatchNormalization(synchronized=True),
    tf.keras.layers.Dense(128,activation = 'relu'),
    tf.keras.layers.BatchNormalization(synchronized=True),
    tf.keras.layers.Dense(10,activation = 'relu'),
    tf.keras.layers.BatchNormalization(synchronized=True),
    tf.keras.layers.Dense(4,activation = 'softmax'),
])
nnmodel.summary()
opt = tf.keras.optimizers.Adam(learning_rate=0.1)
nnmodel.compile(optimizer = opt,loss='categorical_crossentropy',metrics=['accuracy'])
hist = nnmodel.fit(x_train,y_train_cat,batch_size = 21,epochs= 1000,verbose=1)

Epoch 972/1000
172/172 [==============================] - 1s 7ms/step - loss: 0.1586 - accuracy: 0.9144
Epoch 973/1000
172/172 [==============================] - 1s 7ms/step - loss: 0.1552 - accuracy: 0.9182
Epoch 974/1000
172/172 [==============================] - 1s 7ms/step - loss: 0.1823 - accuracy: 0.9088
```

Fig.3.69 Code source second model.

	precision	recall	f1-score	support
0	0.93	0.53	0.68	122
1	0.80	0.98	0.88	252
2	1.00	1.00	1.00	358
3	1.00	0.98	0.99	170
accuracy			0.93	902
macro avg	0.93	0.87	0.89	902
weighted avg	0.94	0.93	0.92	902

Fig.3.70 Code source second model.



```
#test4
input_data = (0.5,21,79,14)
input_data_as_numpy_array = np.asarray(input_data)
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
prediction = nnmodel.predict(input_data_reshaped)
print(crops[np.argmax(prediction)])
```

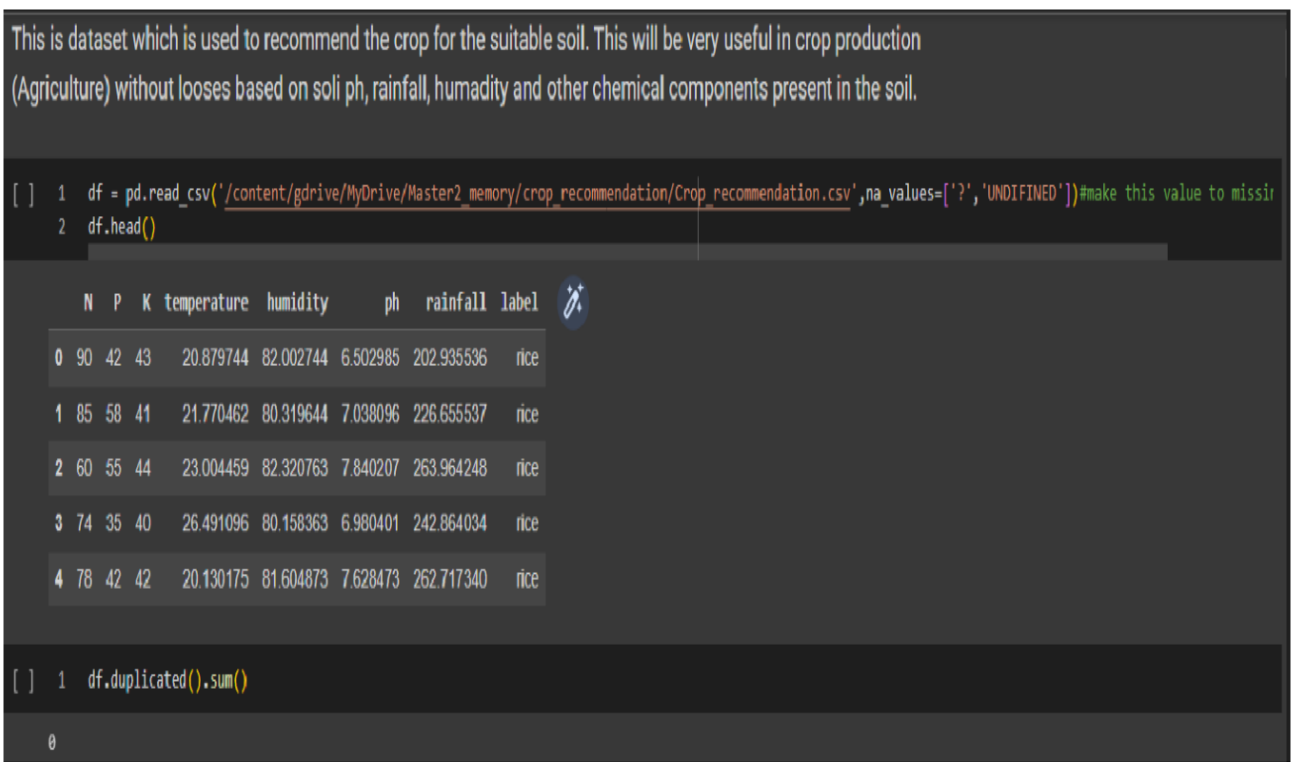
```
1/1 [==============================] - 0s 34ms/step
Rust
```

Fig.3.71 Code source second model.

Discusion

As I see from the previous image my model of ANN obtained good results more than ML algorithms 93%.

Crop recommendation



This is dataset which is used to recommend the crop for the suitable soil. This will be very useful in crop production (Agriculture) without looses based on soli ph, rainfall, humadity and other chemical components present in the soil.

```
df = pd.read_csv('/content/gdrive/MyDrive/Master2_memory/crop_recommendation/Crop_recommendation.csv',na_values=['?','UNDIFINED'])#make this value to missir
df.head()
```

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice

```
df.duplicated().sum()
```

```
0
```

Fig.3.72 Code source of model.

```
# Unique values in the label column

crops = df['label'].unique()
print(len(crops))
print(crops)
print(pd.value_counts(df['label']))
```

```
22
['rice' 'maize' 'chickpea' 'kidneybeans' 'pigeonpeas' 'mothbeans'
 'mungbean' 'blackgram' 'lentil' 'pomegranate' 'banana' 'mango' 'grapes'
 'watermelon' 'muskmelon' 'apple' 'orange' 'papaya' 'coconut' 'cotton'
 'jute' 'coffee']
rice           100
maize          100
jute           100
cotton         100
coconut        100
papaya         100
orange         100
apple          100
muskmelon      100
watermelon     100
grapes         100
```

Fig.3.73 Code source of model.

```
x_train,x_test,y_train,y_test = model_selection.train_test_split(x,y, test_size=0.2, random_state=42)
print("len x train: ",len(x_train),"len y train: ",len(y_train),"len x test: ",len(x_test),"len y test: ",len(y_test))
```

```
len x train:  1760 len y train:  1760 len x test:  440 len y test:  440
```

```
#normalisation for tabular data value between [0,1]

from sklearn.preprocessing import MinMaxScaler
Scaler=MinMaxScaler()
Scaler.fit_transform(x_train)
x_train.head()
#data scaling

```

	N	P	K	temperature	humidity	ph	rainfall
1656	17	16	14	16.396243	92.181519	6.625539	102.944161
752	37	79	19	27.543848	69.347863	7.143943	69.408782
892	7	73	25	27.521856	63.132153	7.288057	45.208411
1041	101	70	48	25.360592	75.031933	6.012697	116.553145

Fig.3.74 Code source of model.

```
#Guassian Naive Bayes
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
NaiveBayes = GaussianNB()
NaiveBayes.fit(x_train,y_train)
y_pred=NaiveBayes.predict(x_test)
ac=accuracy_score(y_test, y_pred)
print("score :",ac*100," %")
acc.append(ac)
model.append("Guassian Naive Bayes")
confusion=confusion_matrix(y_test,y_pred)
plt.figure(figsize=(8,8))
sns.heatmap(confusion,annot=True)
plt.xlabel("Predicted")
plt.ylabel("Actual")
print(classification_report(y_test,y_pred))
```

Fig.3.75 Code source of model.

kidneybeans	1.00	1.00	1.00	20
lentil	1.00	1.00	1.00	11
maize	1.00	1.00	1.00	21
mango	1.00	1.00	1.00	19
mothbeans	1.00	1.00	1.00	24
mungbean	1.00	1.00	1.00	19
muskmelon	1.00	1.00	1.00	17
orange	1.00	1.00	1.00	14
papaya	1.00	1.00	1.00	23
pigeonpeas	1.00	1.00	1.00	23
pomegranate	1.00	1.00	1.00	23
rice	1.00	0.89	0.94	19
watermelon	1.00	1.00	1.00	19
accuracy			1.00	440
macro avg	1.00	1.00	1.00	440
weighted avg	1.00	1.00	1.00	440

Fig.3.76 Code source of model.

```python
pd.DataFrame(acc,model)
```

	0
KNeighbors Classifier	0.965909
Support Vector Classifier	0.977273
Logistic Regression	0.954545
Random Forest Classifier	0.993182
Gradient Boosting Classifier	0.984091
LGBM Classifier	0.986364
Decision TreeClassifier	0.865909
Guassian Naive Bayes	0.995455
Support Vector Machine (SVM)	0.090909
Extra Tree Classifier	0.988636
AdaBoost	0.140909

Fig.3.77 Code source of model.

```python
import tensorflow as tf
import pickle
#new_model = tf.keras.models.load_model('/content/gdrive/MyDrive/Master2_memory/crop_recommendation/NaiveBayes.pkl')

# Check its architecture
#new_model.summary()
pickled_model = pickle.load(open('/content/gdrive/MyDrive/Master2_memory/crop_recommendation/NaiveBayes.pkl', 'rb'))
#pickled_model.predict(X_test)
```

```python
# Define function to make predictions
def predict_crop(N, P, K, temperature, humidity, pH, rainfall):
    # Create a numpy array with the input values
    input_values = np.array([[N, P, K, temperature, humidity, pH, rainfall]])

    # Use the model to make a prediction
    prediction = pickled_model.predict(input_values)

    # Return the predicted crop label
    return prediction[0]
```

Fig.3.78 Code source of model.

```python
N = 90
P = 42
K = 43
tem = 20.87974371
humidity = 82.00274423
ph = 6.502985292
rainfall = 202.9355362

pred = predict_crop(N, P, K, tem, humidity, ph, rainfall)
print(pred)
```

```
rice
```

Fig.3.79 Code source of model.

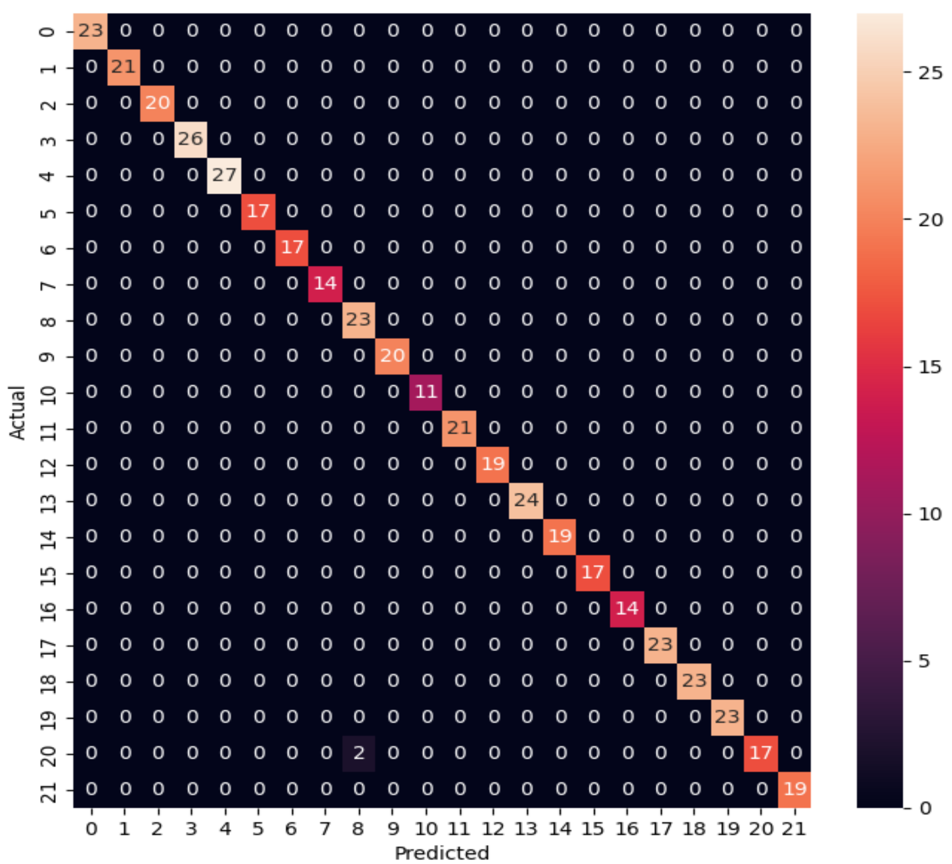


Fig.3.80 Code source of model.

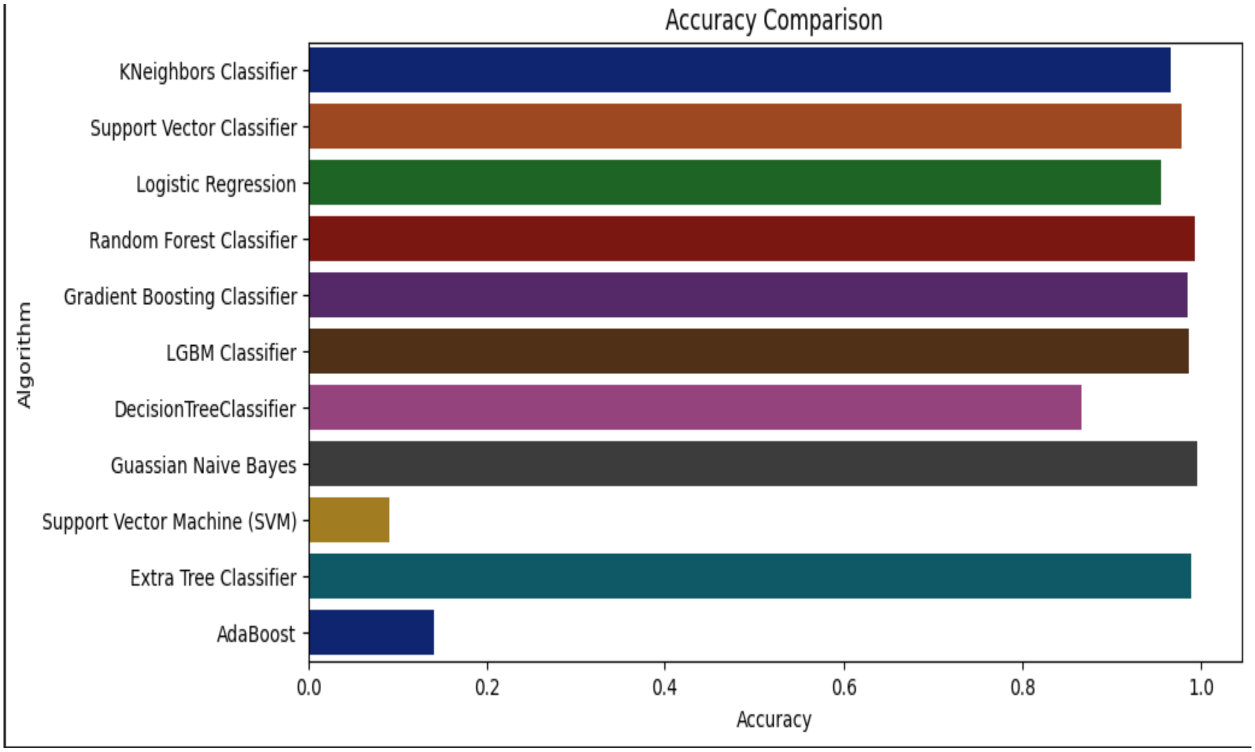


Fig.3.81 Code source of model.

Discusion

The best result is obtained by Naives Bayes classifier.99.54%.

Fertilizer prediction

```
[ ]  print("Fertilizer_Name    :   \n",df['Fertilizer_Name'].unique())

Fertilizer_Name    :
 ['Urea' 'DAP' '14-35-14' '28-28' '17-17-17' '20-20' '10-26-26']
```

```
[ ]  print("number of data for each Fertilizer_Name :   \n ",df['Fertilizer_Name'].value_counts())

number of data for each Fertilizer_Name :
  Urea        44
DAP         38
28-28       36
20-20       30
14-35-14    29
17-17-17    18
10-26-26    10
Name: Fertilizer_Name, dtype: int64
```

```
[ ]  print("columns:   \n",df.columns)

columns:
 Index(['Temparature', 'Humidity ', 'Moisture', 'Soil_Type', 'Crop Type',
       'Nitrogen', 'Potassium', 'Phosphorous', 'Fertilizer_Name'],
      dtype='object')
```

Fig.3.82 Code source of model.

```
#data_encoded
df=df.replace({
    'Soil_Type':{'Sandy':0,'Loamy':1, 'Black':2,'Red':3,'Clayey':4},
    'Crop Type':{'Maize':0,'Sugarcane':1, 'Cotton':2,'Tobacco':3,'Paddy':4,'Barley':5,'Wheat':6,'Millets':7,
                'Oil seeds':8,'Pulses':8,'Ground Nuts':9,},
})
df.head()
```

Fig.3.83 Code source of model.

```
#normalisation for tabular data value between [0,1]
from sklearn.preprocessing import MinMaxScaler
Scaler=MinMaxScaler()
Scaler.fit_transform(x_train)
x_train.head()
```

1 to 5 of 5 entries Filter

index	Temparature	Humidity	Moisture	Soil_Type	Crop Type	Nitrogen	Potassium	Phosphorous
99	36	68	38	0	5	7	9	30
143	31	62	49	2	1	10	13	14
21	34	65	53	1	1	12	14	12
28	37	70	32	2	8	12	0	39
160	25	50	64	3	2	9	0	10

Show 25 per page

Fig.3.84 Code source of model.

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score
# Initializing empty lists to append all model's name and corresponding name
acc = []
model = []
knn=KNeighborsClassifier(n_neighbors=1)
knn.fit(x_train,y_train)
from sklearn.metrics import confusion_matrix
ac=accuracy_score(y_test, knn.predict(x_test))
print("score :",ac*100," %")
acc.append(ac)
model.append("KNeighbors Classifier")
confusion_knn=confusion_matrix(y_test,knn.predict(x_test))
plt.figure(figsize=(8,8))
sns.heatmap(confusion_knn,annot=True)
plt.xlabel("Predicted")
plt.ylabel("Actual")
from sklearn.metrics import classification_report
print(classification_report(y_test,knn.predict(x_test)))
```

Fig.3.85 Code source of model.

```
score : 100.0  %
              precision    recall  f1-score   support

    10-26-26       1.00      1.00      1.00         2
    14-35-14       1.00      1.00      1.00         6
    17-17-17       1.00      1.00      1.00         4
       20-20       1.00      1.00      1.00         6
       28-28       1.00      1.00      1.00         7
         DAP       1.00      1.00      1.00         7
        Urea       1.00      1.00      1.00         9

    accuracy                           1.00        41
   macro avg       1.00      1.00      1.00        41
weighted avg       1.00      1.00      1.00        41
```

Fig.3.86 Code source of model.

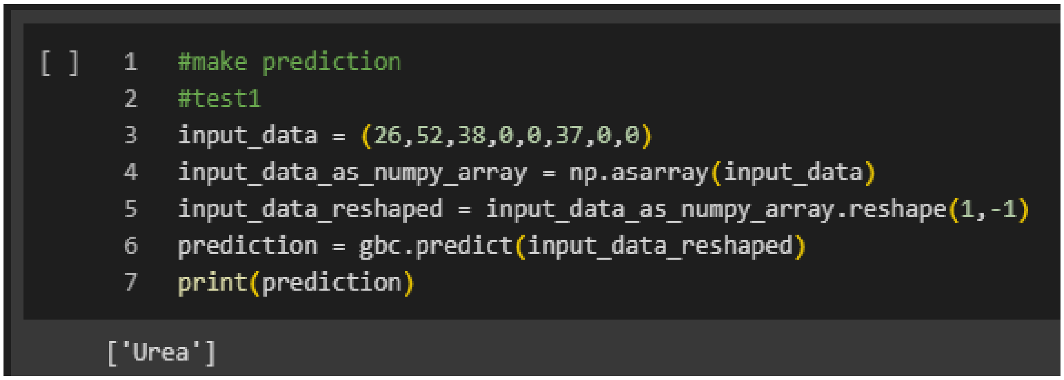
```
1   pd.DataFrame(acc,model)
```

1 to 11 of 11 entries Filter

index	0
KNeighbors Classifier	1.0
Support Vector Classifier	1.0
Logistic Regression	0.975609756097561
Random Forest Classifier	0.975609756097561
Gradient Boosting Classifier	1.0
LGBM Classifier	1.0
DecisionTreeClassifier	1.0
Guassian Naive Bayes	0.975609756097561
Support Vector Machine (SVM)	0.8780487804878049
Extra Tree Classifier	1.0
AdaBoost	0.6097560975609756

Show 25 per page

Fig.3.87 Code source of model.



```
[ ]
#make prediction
#test1
input_data = (26,52,38,0,0,37,0,0)
input_data_as_numpy_array = np.asarray(input_data)
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
prediction = gbc.predict(input_data_reshaped)
print(prediction)

['Urea']
```

Fig.3.88 Code source of model.

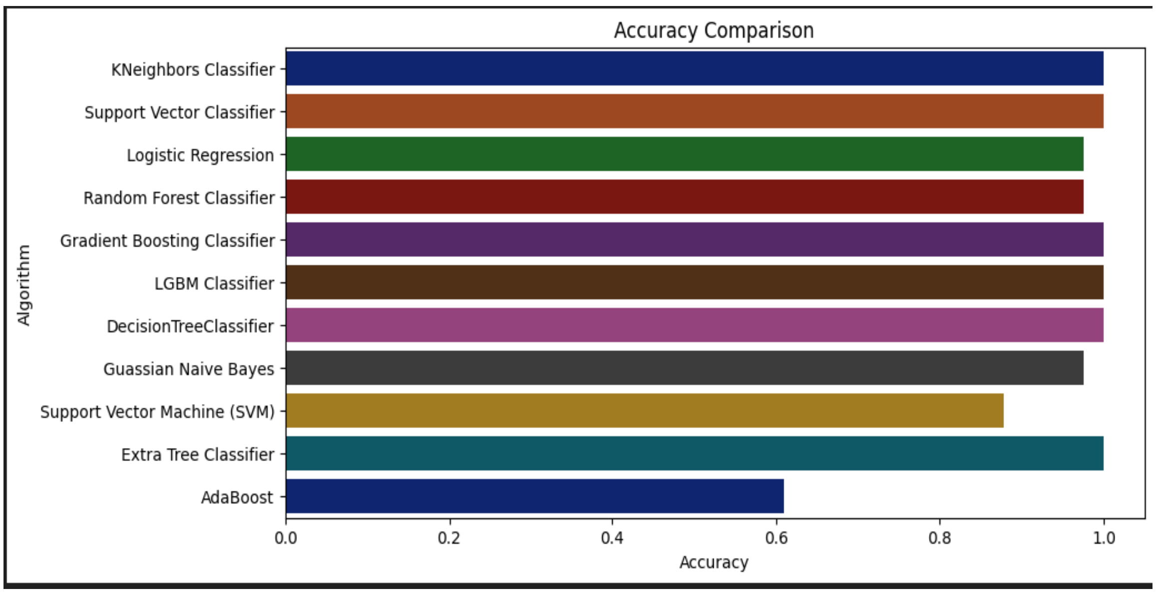


Fig.3.89 Code source of model.

Discusion

The best result is obtained by K neighbor classifier 100%.

Pest prediction

I trained the same model of CNN with pest dataset.

```
- accuracy: 1.0000 - categorical_accuracy: 1.0000 - precision_1: 1.0000 - recall_1: 1.0000 - val_loss: 1.4705 - val_
- accuracy: 1.0000 - categorical_accuracy: 1.0000 - precision_1: 1.0000 - recall_1: 1.0000 - val_loss: 1.4692 - val_accuracy: 0.6203 -
- accuracy: 1.0000 - categorical_accuracy: 1.0000 - precision_1: 1.0000 - recall_1: 1.0000 - val_loss: 1.4734 - val_accuracy: 0.6250 -
- accuracy: 1.0000 - categorical_accuracy: 1.0000 - precision_1: 1.0000 - recall_1: 1.0000 - val_loss: 1.4817 - val_accuracy: 0.6231 -
- accuracy: 1.0000 - categorical_accuracy: 1.0000 - precision_1: 1.0000 - recall_1: 1.0000 - val_loss: 1.4806 - val_accuracy: 0.6222 -
- accuracy: 1.0000 - categorical_accuracy: 1.0000 - precision_1: 1.0000 - recall_1: 1.0000 - val_loss: 1.4857 - val_accuracy: 0.6269 -
- accuracy: 1.0000 - categorical_accuracy: 1.0000 - precision_1: 1.0000 - recall_1: 1.0000 - val_loss: 1.4932 - val_accuracy: 0.6222 -
- accuracy: 1.0000 - categorical_accuracy: 1.0000 - precision_1: 1.0000 - recall_1: 1.0000 - val_loss: 1.4892 - val_accuracy: 0.6222 -
- accuracy: 1.0000 - categorical_accuracy: 1.0000 - precision_1: 1.0000 - recall_1: 1.0000 - val_loss: 1.4889 - val_accuracy: 0.6259 -
- accuracy: 1.0000 - categorical_accuracy: 1.0000 - precision_1: 1.0000 - recall_1: 1.0000 - val_loss: 1.4849 - val_accuracy: 0.6231 -
```

Fig.3.90 Code source of model.

```
def show_image_samples(gen ):
    t_dict=gen.class_indices
    classes=list(t_dict.keys())
    images,labels=next(gen) # get a sample batch from the generator
    plt.figure(figsize=(25, 25))
    length=len(labels)
    if length<25:   #show maximum of 25 images
        r=length
    else:
        r=25
    for i in range(r):
        plt.subplot(5, 5, i + 1)
        image=images[i] /255
        plt.imshow(image)
        index=np.argmax(labels[i])
        class_name=classes[index]
        plt.title(class_name, color='blue', fontsize=18)
        plt.axis('off')
    plt.show()

show_image_samples(train_gen )
```

Fig.3.91 Code source of model.

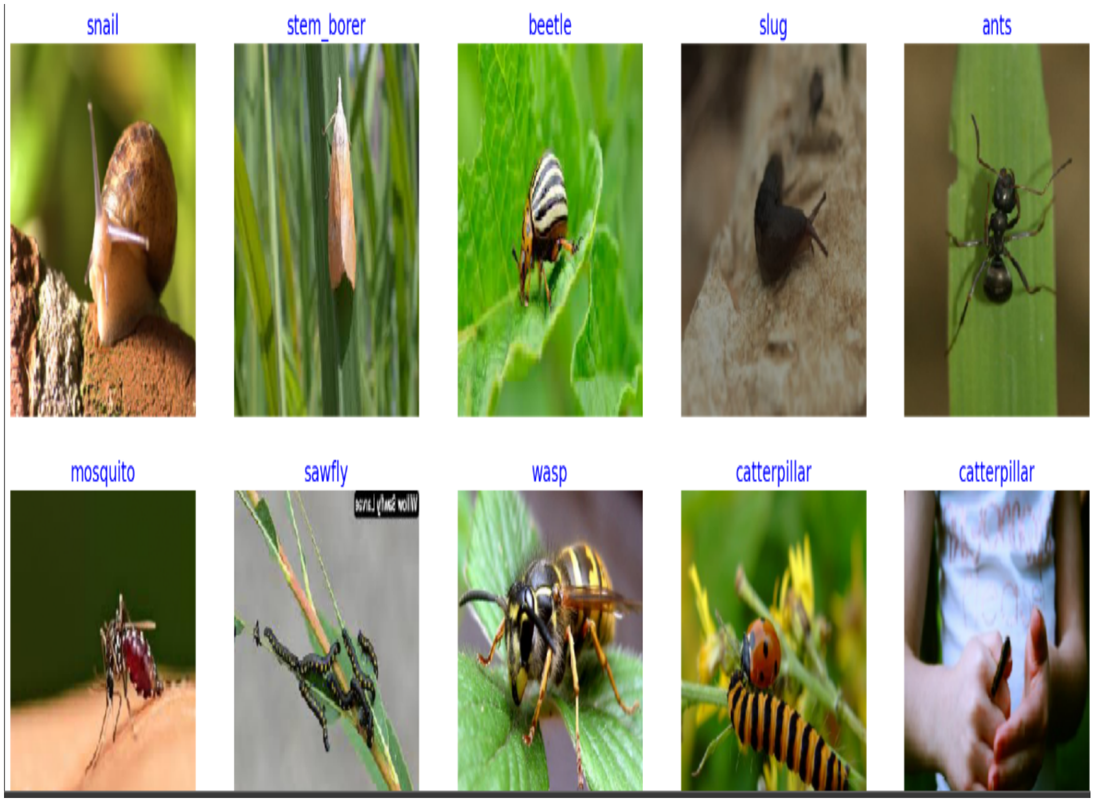


Fig.3.92 Code source of model.

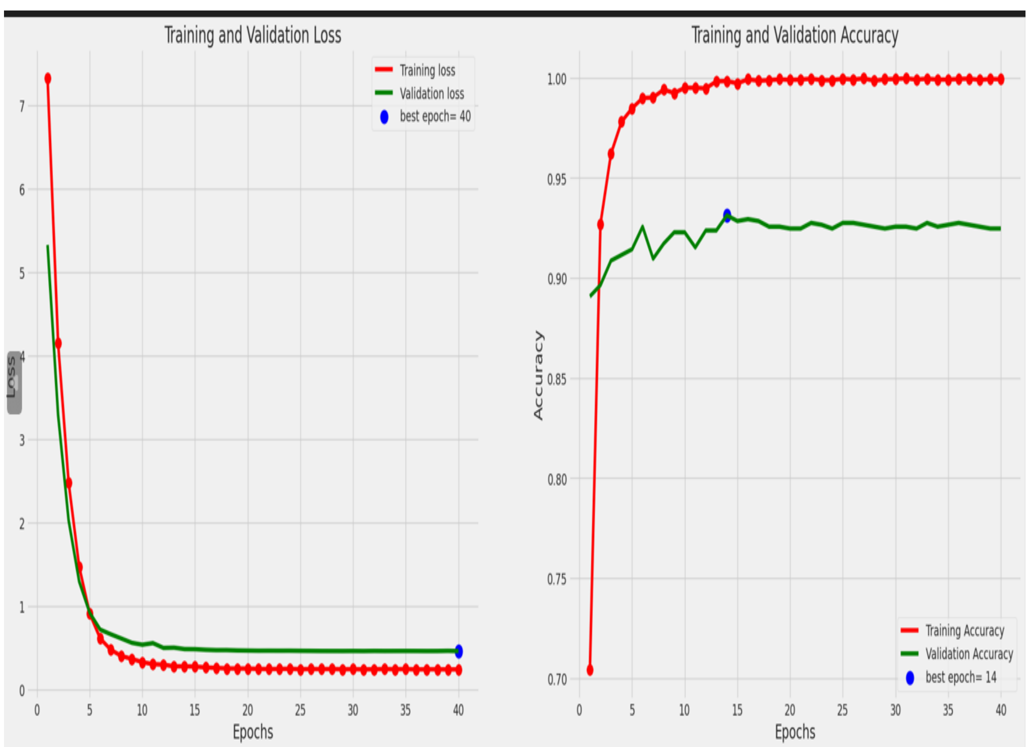


Fig.3.93 Code source of model.

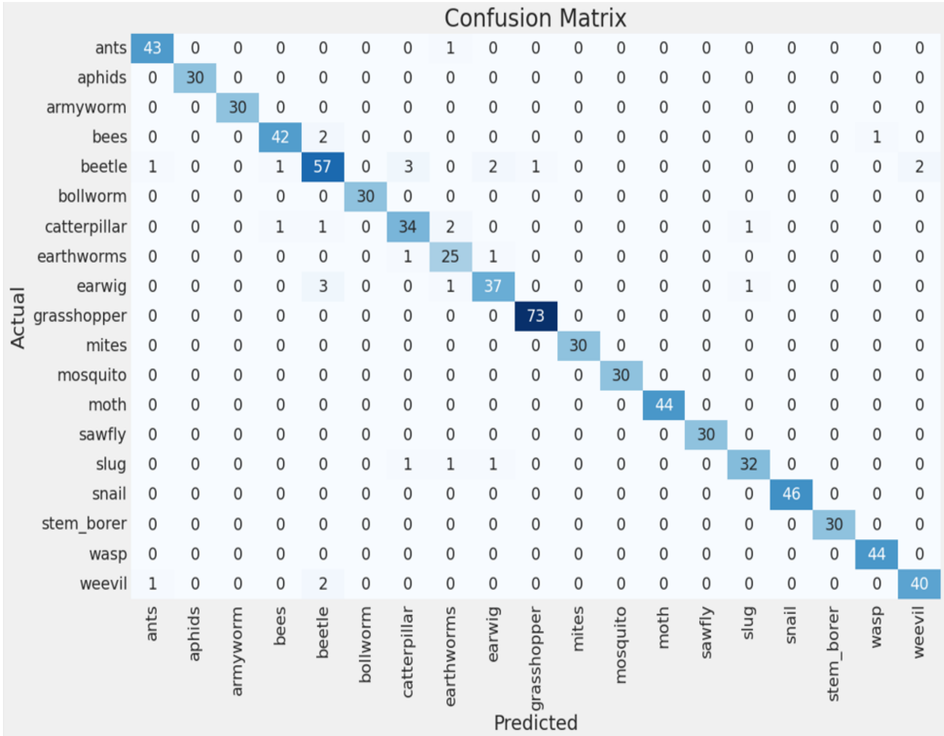
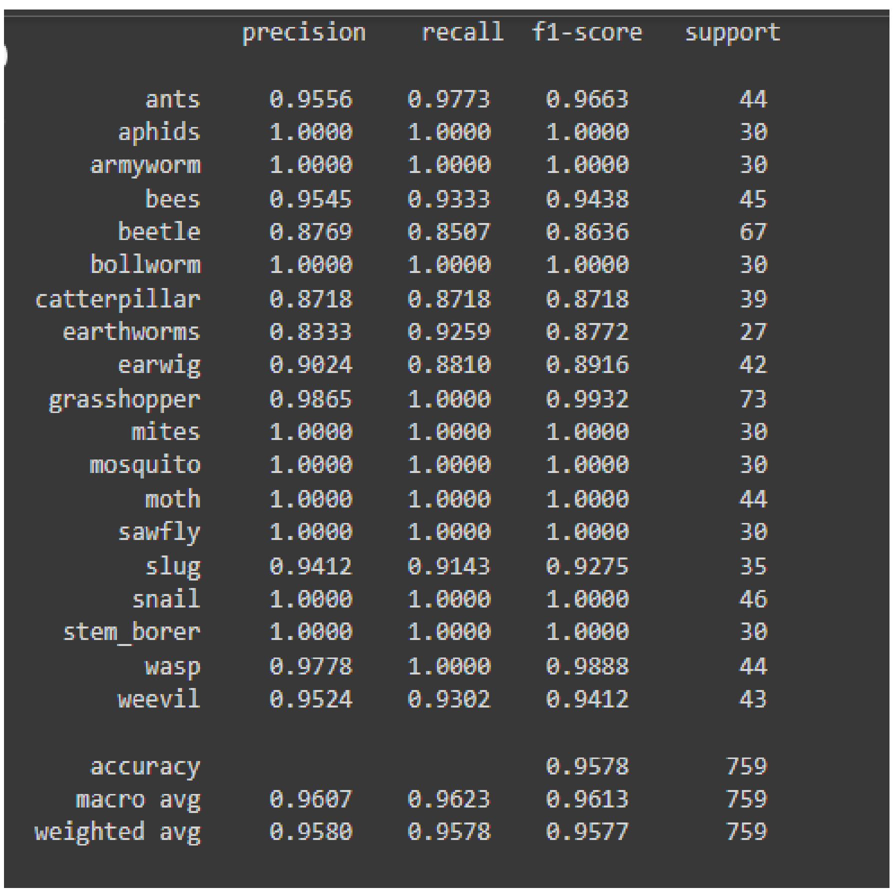


Fig.3.94 Code source of model.



	precision	recall	f1-score	support
ants	0.9556	0.9773	0.9663	44
aphids	1.0000	1.0000	1.0000	30
armyworm	1.0000	1.0000	1.0000	30
bees	0.9545	0.9333	0.9438	45
beetle	0.8769	0.8507	0.8636	67
bollworm	1.0000	1.0000	1.0000	30
catterpillar	0.8718	0.8718	0.8718	39
earthworms	0.8333	0.9259	0.8772	27
earwig	0.9024	0.8810	0.8916	42
grasshopper	0.9865	1.0000	0.9932	73
mites	1.0000	1.0000	1.0000	30
mosquito	1.0000	1.0000	1.0000	30
moth	1.0000	1.0000	1.0000	44
sawfly	1.0000	1.0000	1.0000	30
slug	0.9412	0.9143	0.9275	35
snail	1.0000	1.0000	1.0000	46
stem_borer	1.0000	1.0000	1.0000	30
wasp	0.9778	1.0000	0.9888	44
weevil	0.9524	0.9302	0.9412	43
accuracy			0.9578	759
macro avg	0.9607	0.9623	0.9613	759
weighted avg	0.9580	0.9578	0.9577	759

Fig.3.95 Code source of model.

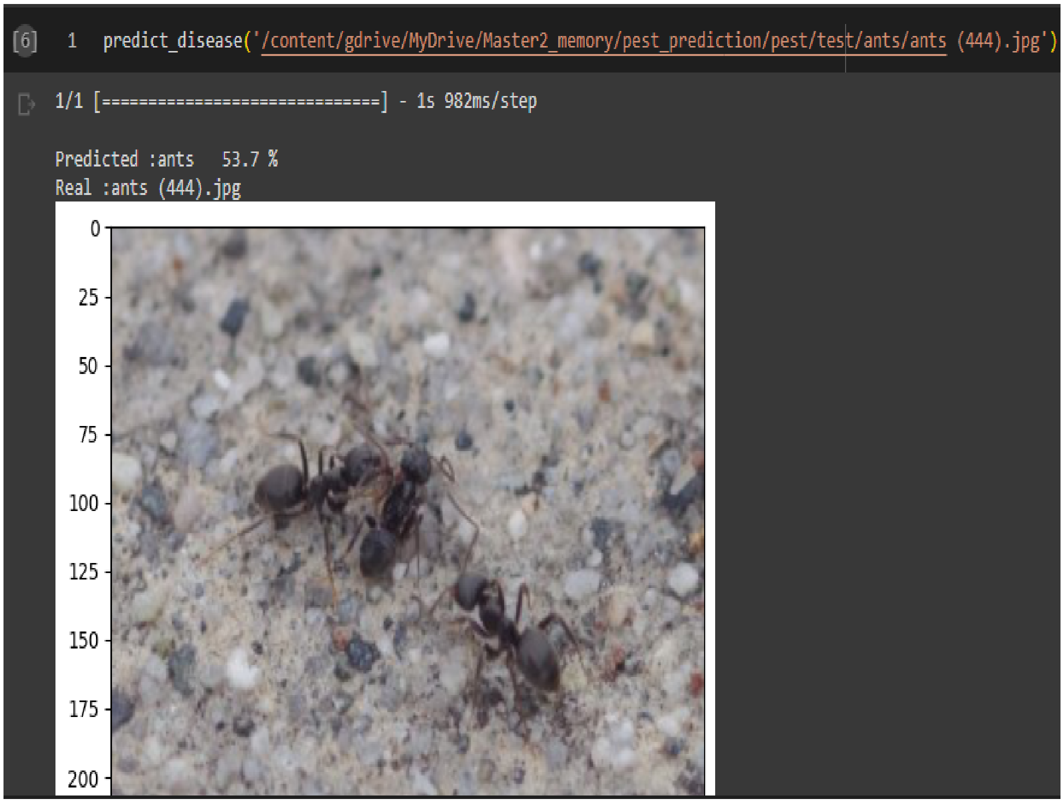


Fig.3.96 Code source of model.

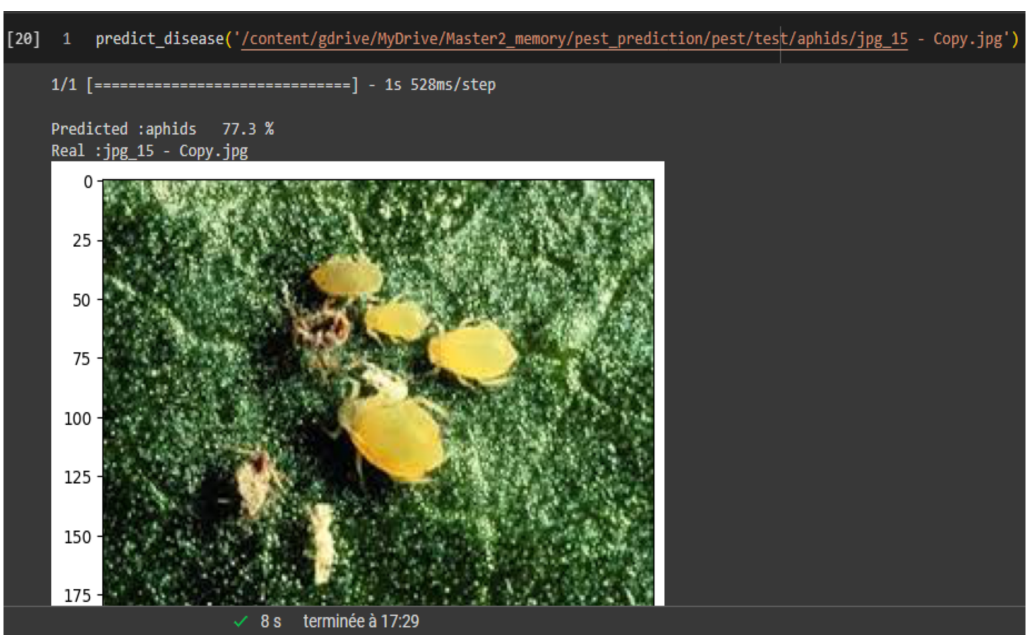


Fig.3.97 Code source of model.

Discusion : I obtained good result in pest detection 95.75% in test.

Weed prediction

```
[23]:
train_datagen = tf.keras.preprocessing.image.ImageDataGenerator(
    rescale = 1./255,
    zoom_range = 0.2,
    height_shift_range = 0.2,
    width_shift_range = 0.2,
    horizontal_flip = True,
    validation_split = 0.2
)
test_datagen = tf.keras.preprocessing.image.ImageDataGenerator(rescale = 1./255)
```

Fig.3.98 Code source of model.

```
8]:
basemodel = tf.keras.applications.ResNet152V2(
    weights = "imagenet",
    input_shape = (img_rows, img_cols, 3),
    include_top = False
)
for layer in basemodel.layers:
    layer.trainable = False
x = basemodel.output
x = tf.keras.layers.GlobalAveragePooling2D()(x)
x = tf.keras.layers.Dense(1024, activation="relu")(x)
x = tf.keras.layers.Dense(1024, activation="relu")(x)
output = tf.keras.layers.Dense(classes, activation="softmax")(x)
model = tf.keras.Model(inputs = basemodel.inputs, outputs = output)
model.summary()
```

```
Model: "model_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_2 (InputLayer)            [(None, 224, 224, 3) 0
__________________________________________________________________________________________________
conv1_pad (ZeroPadding2D)       (None, 230, 230, 3)  0           input_2[0][0]
__________________________________________________________________________________________________
conv1_conv (Conv2D)             (None, 112, 112, 64) 9472        conv1_pad[0][0]
__________________________________________________________________________________________________
pool1_pad (ZeroPadding2D)       (None, 114, 114, 64) 0           conv1_conv[0][0]
__________________________________________________________________________________________________
pool1_pool (MaxPooling2D)       (None, 56, 56, 64)   0           pool1_pad[0][0]
__________________________________________________________________________________________________
conv2_block1_preact_bn (BatchNo (None, 56, 56, 64)   256         pool1_pool[0][0]
```

Fig.3.99 Code source of model.

```
model.compile(
    loss = "categorical_crossentropy",
    optimizer = tf.keras.optimizers.Adam(learning_rate = 0.001),
    metrics = ['accuracy']
)
```

```
history = model.fit(
    train_generator,
    epochs = 20,
    validation_data = val_generator
)
```

```
Epoch 1/20
194/194 [==============================] - 147s 760ms/step - loss: 0.6291 - accuracy: 0.7729 - val_loss: 0.5667 - val_accuracy: 0.7929
Epoch 2/20
194/194 [==============================] - 146s 751ms/step - loss: 0.5302 - accuracy: 0.8090 - val_loss: 0.5640 - val_accuracy: 0.7890
Epoch 3/20
194/194 [==============================] - 148s 766ms/step - loss: 0.4881 - accuracy: 0.8224 - val_loss: 0.6002 - val_accuracy: 0.7910
Epoch 4/20
194/194 [==============================] - 146s 752ms/step - loss: 0.4354 - accuracy: 0.8383 - val_loss: 0.4656 - val_accuracy: 0.8303
Epoch 5/20
194/194 [==============================] - 146s 755ms/step - loss: 0.4190 - accuracy: 0.8454 - val_loss: 0.4216 - val_accuracy: 0.8452
Epoch 6/20
```

Fig.3.100 Code source of model.

```
Epoch 16/20
194/194 [==============================] - 147s 757ms/step - loss: 0.2733 - accuracy: 0.9039 - val_loss: 0.3474 - val_accuracy: 0.8832
Epoch 17/20
194/194 [==============================] - 146s 754ms/step - loss: 0.2714 - accuracy: 0.9017 - val_loss: 0.3890 - val_accuracy: 0.8626
Epoch 18/20
194/194 [==============================] - 146s 755ms/step - loss: 0.2562 - accuracy: 0.9112 - val_loss: 0.3548 - val_accuracy: 0.8832
Epoch 19/20
194/194 [==============================] - 148s 762ms/step - loss: 0.2551 - accuracy: 0.9073 - val_loss: 0.3945 - val_accuracy: 0.8748
Epoch 20/20
194/194 [==============================] - 146s 752ms/step - loss: 0.2621 - accuracy: 0.9026 - val_loss: 0.3439 - val_accuracy: 0.8794
```

```
[32]:
scores = model.evaluate(val_generator, verbose = 1)
print("Loss: {:.3f}".format(scores[0]))
print("Accuracy: {:.3f}".format(scores[1]))
```

```
49/49 [==============================] - 30s 605ms/step - loss: 0.3602 - accuracy: 0.8761
Loss: 0.360
Accuracy: 0.876
```

```
[33]:
model.save('/kaggle/working/weed_prediction.h5')
```

Fig.3.101 Code source of model.

```
5]:
class_labels = test_generator.class_indices
class_labels = {v:k for k,v in class_labels.items()}
classes = list(class_labels.values())
classes
```

```
5_ ['Black-grass',
 'Charlock',
 'Cleavers',
 'Common Chickweed',
 'Common wheat',
 'Fat Hen',
 'Loose Silky-bent',
 'Maize',
 'Scentless Mayweed',
 'Shepherd's Purse',
 'Small-flowered Cranesbill',
 'Sugar beet']
```

Fig.3.102 Code source of model.

```
37]:
print("Classification Report")
print(classification_report(test_generator.classes, test_labels,target_names = classes))
```

Classification Report

	precision	recall	f1-score	support
Black-grass	0.65	0.69	0.67	189
Charlock	1.00	0.96	0.98	278
Cleavers	0.98	0.93	0.95	193
Common Chickweed	0.92	0.93	0.93	407
Common wheat	0.85	0.93	0.89	148
Fat Hen	0.88	0.93	0.91	348
Loose Silky-bent	0.88	0.85	0.86	480
Maize	0.91	0.99	0.95	141
Scentless Mayweed	0.93	0.90	0.92	366
Shepherd's Purse	0.88	0.75	0.81	155
Small-flowered Cranesbill	0.92	0.99	0.95	345
Sugar beet	0.95	0.89	0.92	274
accuracy			0.90	3324
macro avg	0.90	0.90	0.89	3324
weighted avg	0.90	0.90	0.90	3324

Fig.3.103 Code source of model.

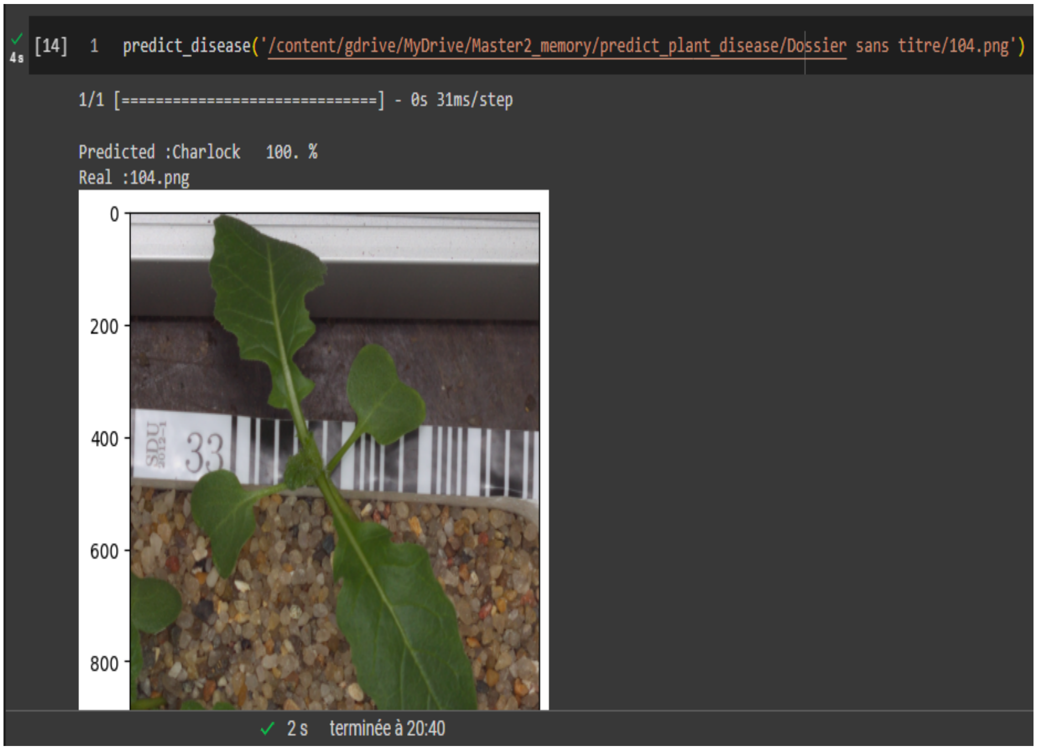


Fig.3.104 Code source of model.

Discusion : I obtained good ,result in weed detection 90% test.

My Web site and mobile application

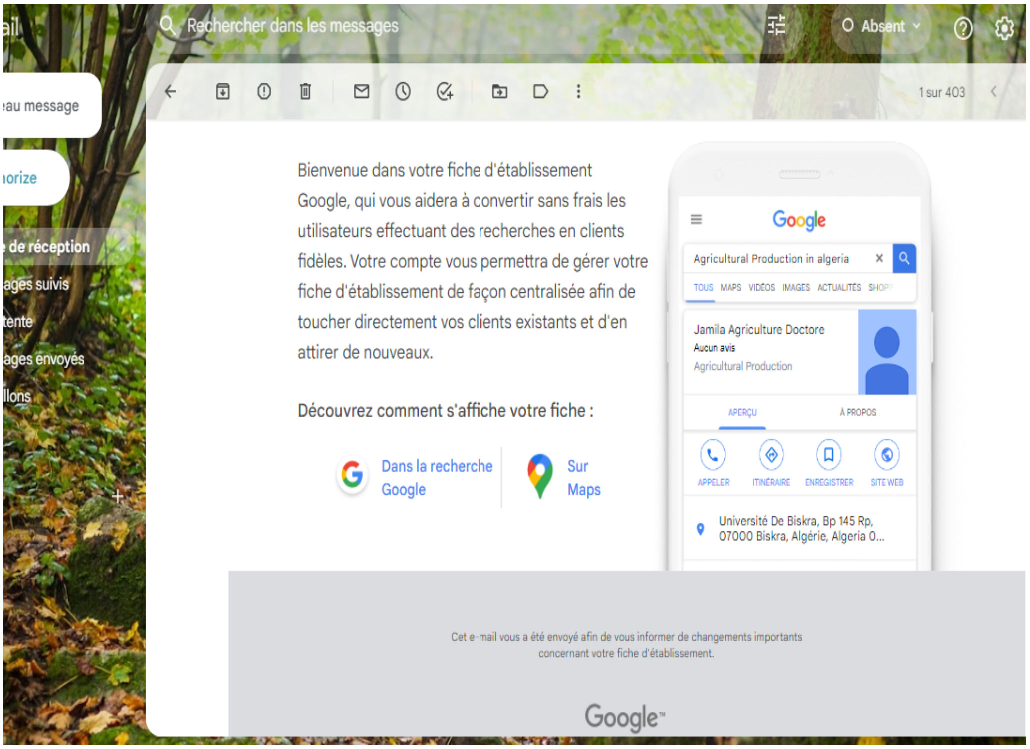


Fig.4.My web site visible in google.

Fig.4.1 My mobile application.

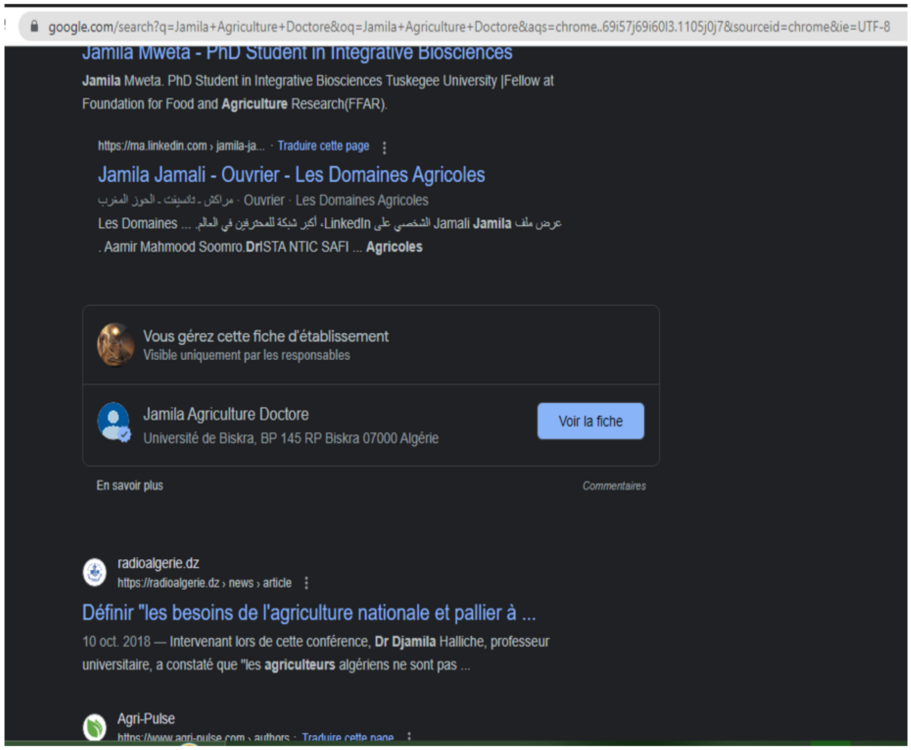


Fig.4.2 My web site visible in google.

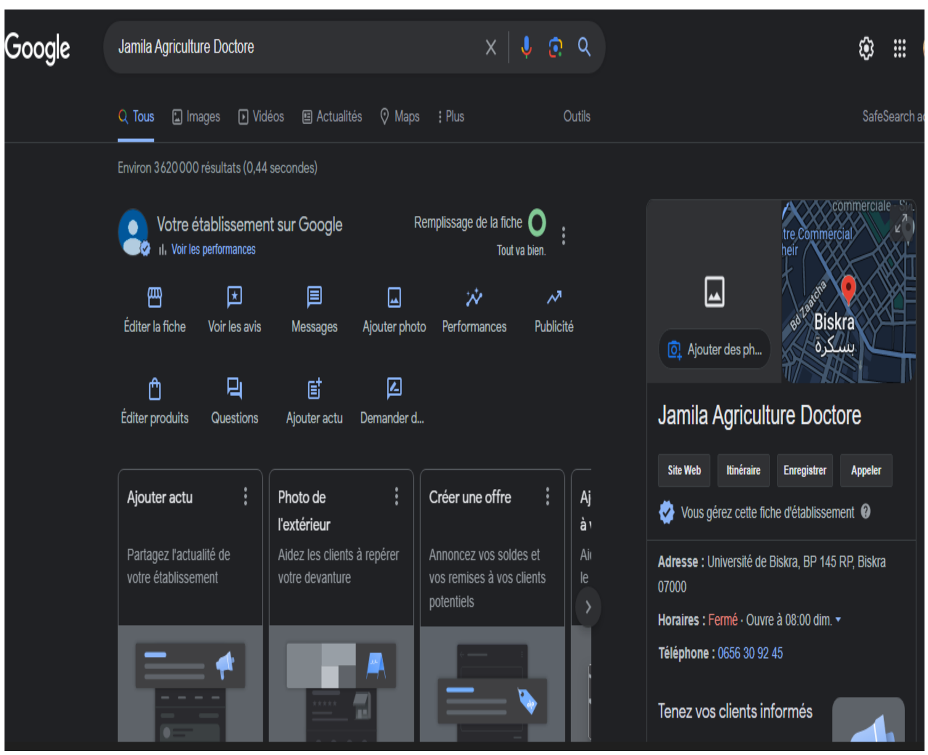


Fig.4.3 My web site visible in google.

My web site : is adaptable to mobile

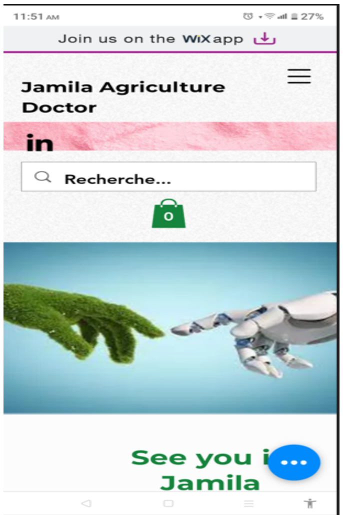


Fig.4.4 My web site visible in google.

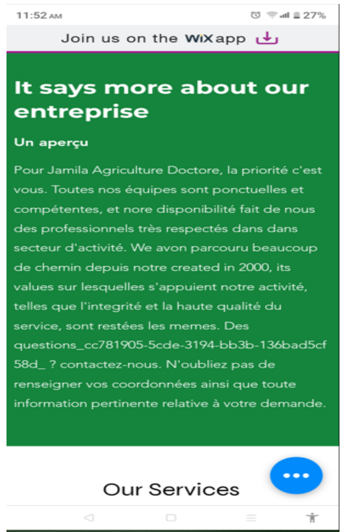


Fig.4.5 My web site visible in google.

Fig.4.6 My web site visible in google.

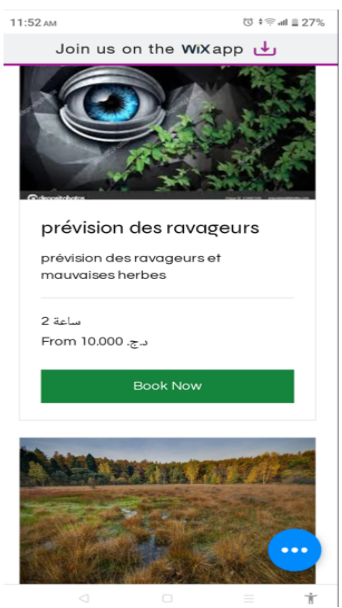


Fig.4.7 My web site visible in google.

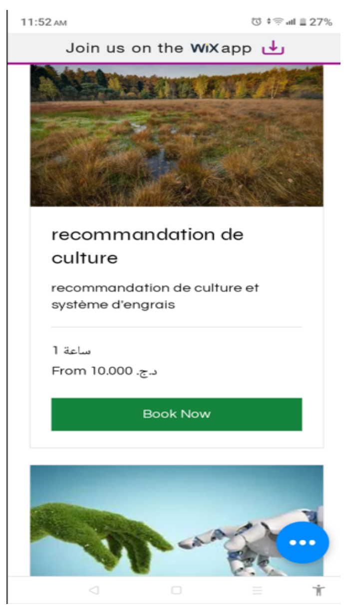


Fig.4.8 My web site visible in google.

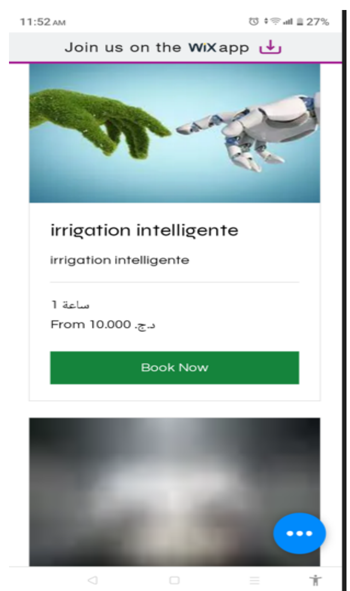


Fig.4.9 My web site visible in google.

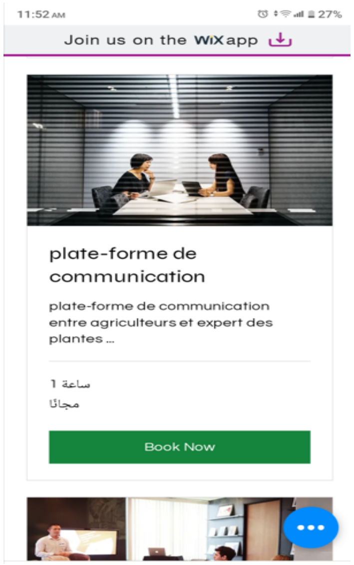


Fig.4.10 My web site visible in google.

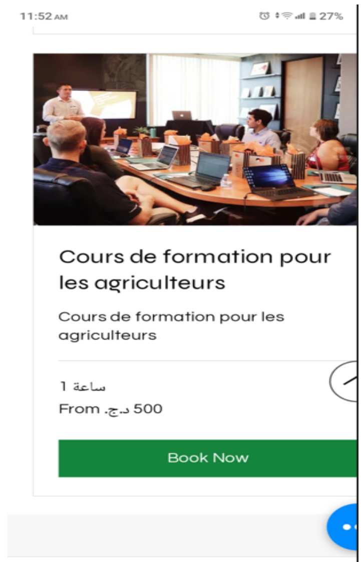


Fig.4.11 My web site visible in google.

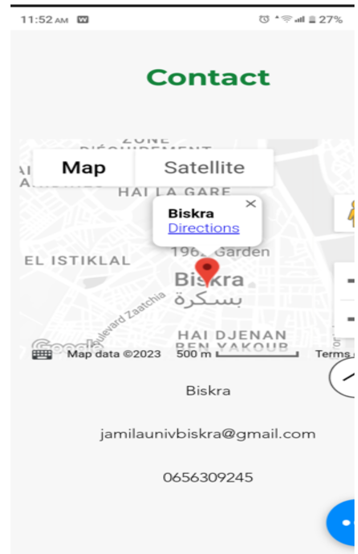


Fig.4.12 My web site visible in google.

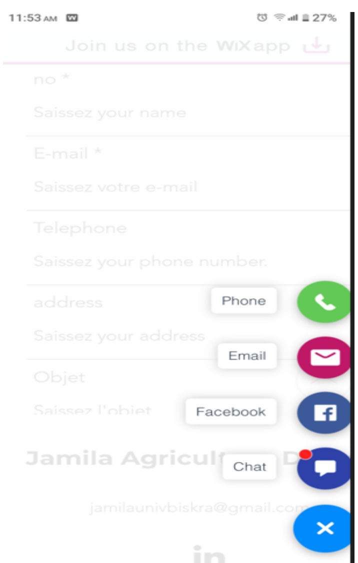


Fig.4.13 My web site visible in google.

Fig.4.14 My web site visible in google.

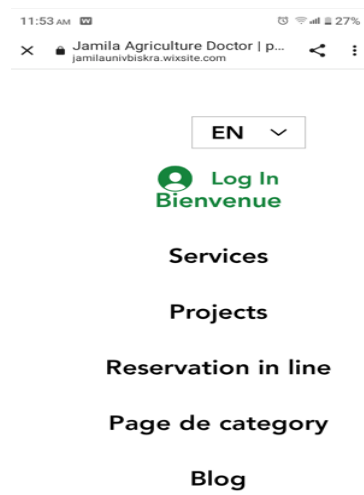


Fig.4.15 My web site visible in google.

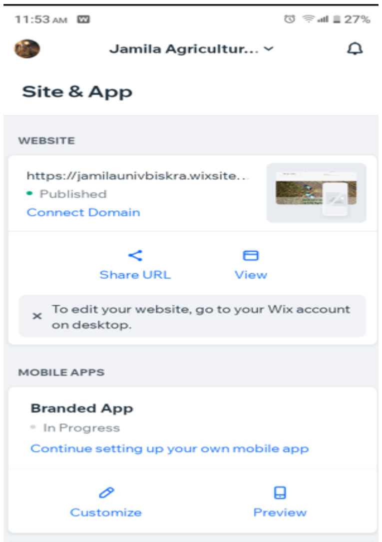


Fig.4.16 My web site visible in google.

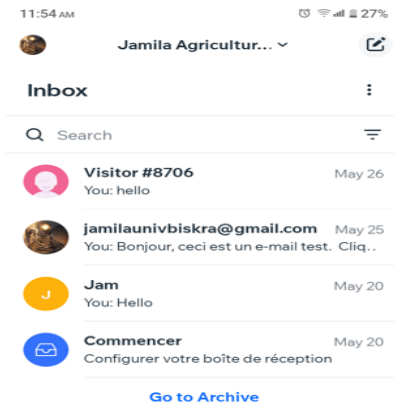
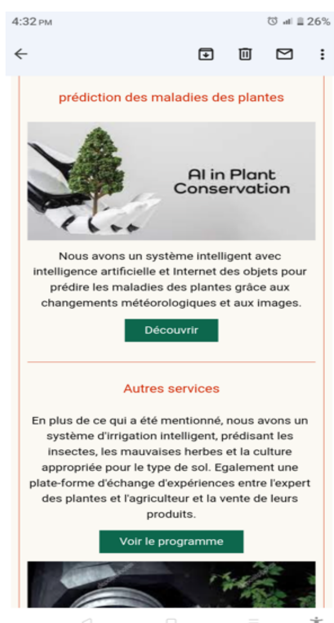


Fig.4.17 My web site visible in google.

The email send it to user subscribed :



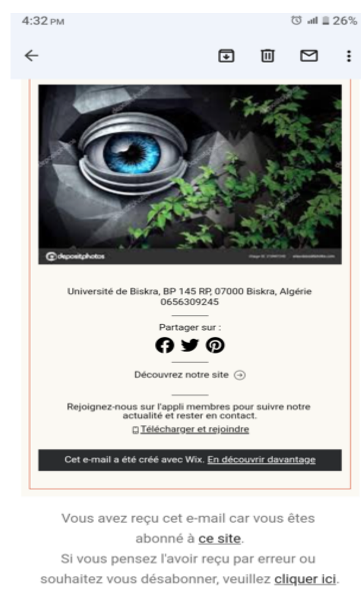


Fig.4.18 The email send it to user subscribed

Fig.4.19 The email send it to user subscribed

Fig.4.20 The email send it to user subscribed

My mobile application :

Fig.5 My mobile application

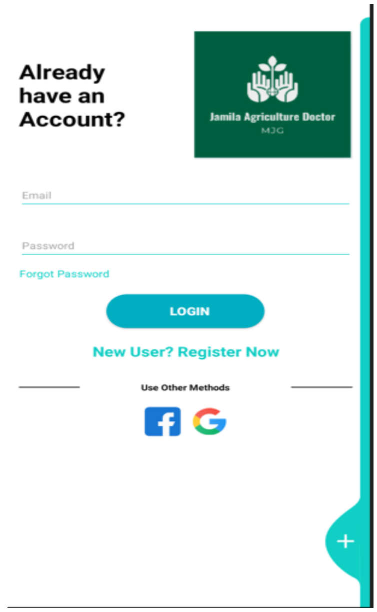


Fig.5.1 My mobile application

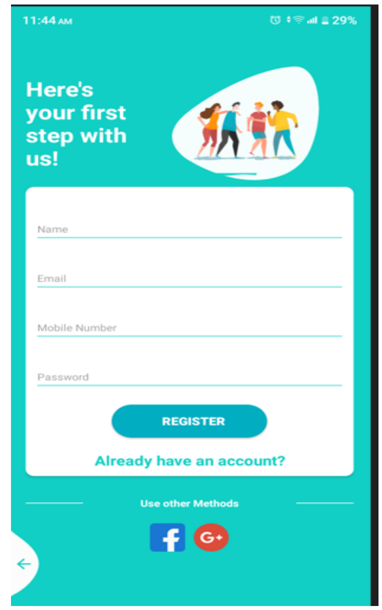


Fig.5.2 My mobile application

Fig.5.3 My mobile application

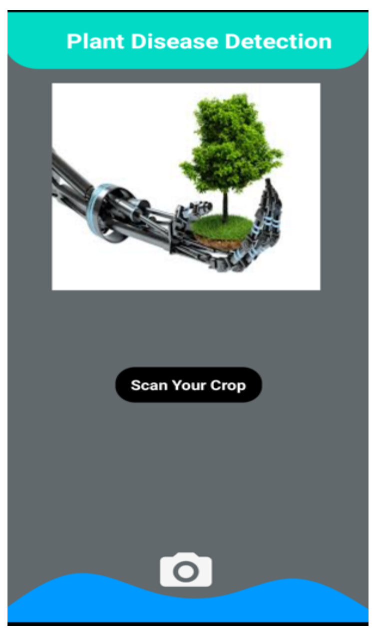


Fig.5.4 My mobile application

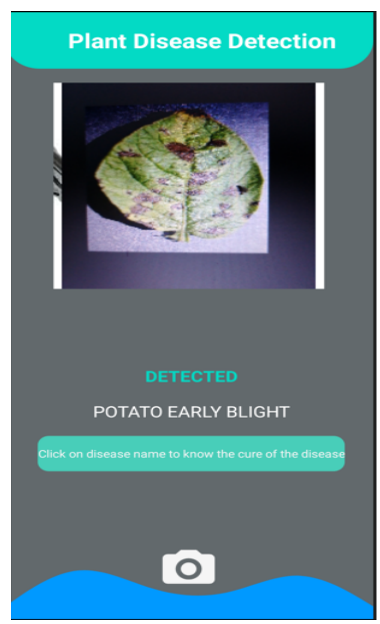


Fig.5.5 My mobile application

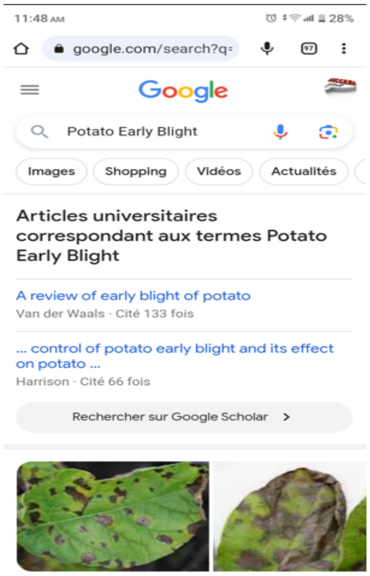


Fig.5.6 My mobile application

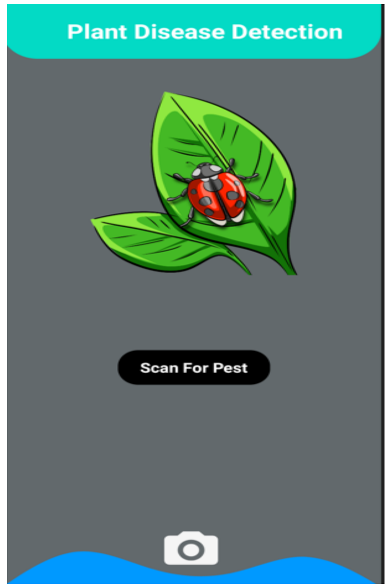


Fig.5.7 My mobile application

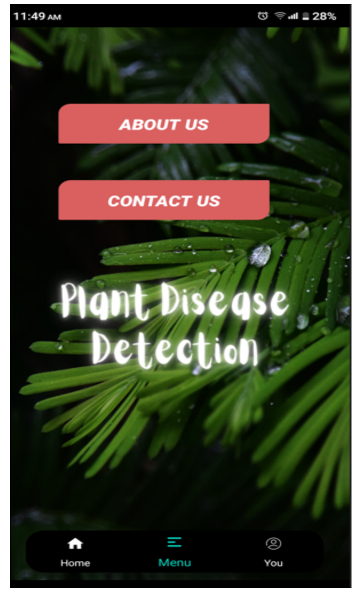


Fig.5.8 My mobile application

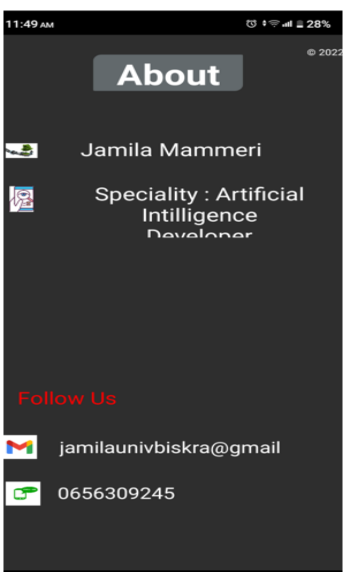


Fig.5.9 My mobile application

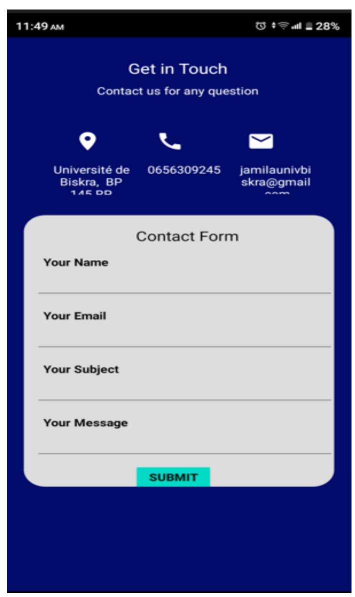


Fig.5.10 My mobile application

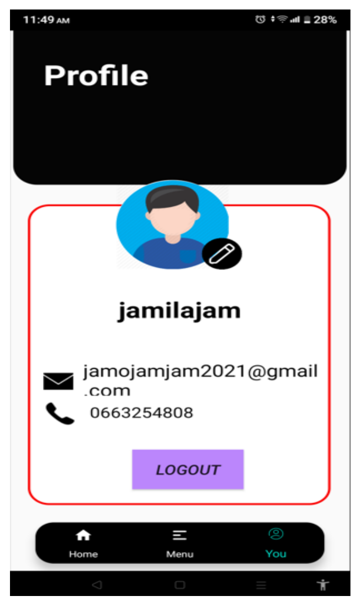


Fig.5.11 My mobile application

Fig.5.12 My mobile application to communication between farmers.

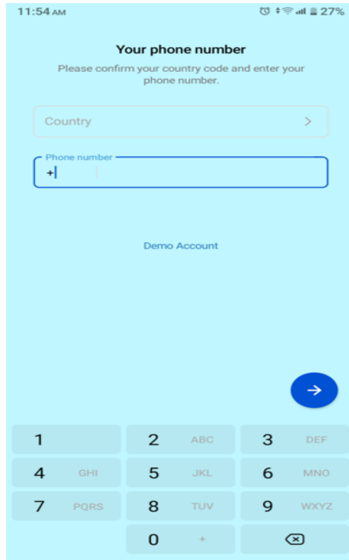


Fig.5.13 My mobile application to communication between farmers.

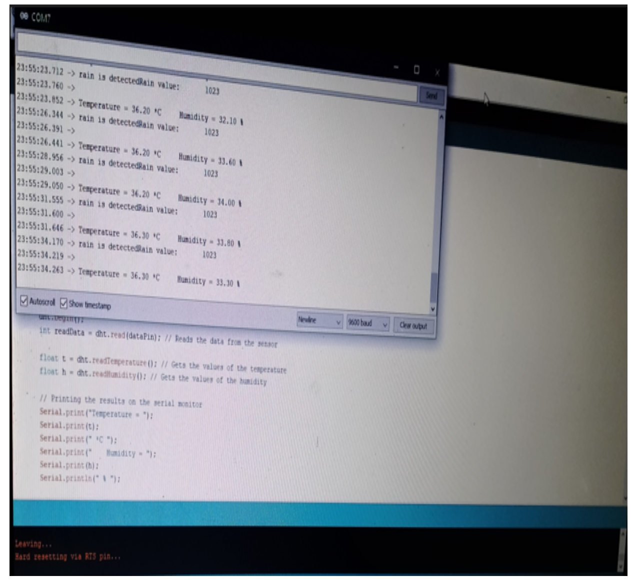


Fig.5.14 The result of DHT22 and rainfall.

My web site

check out my website,

"Jamila Agriculture D" with this link : https ://jamilaunivbiskra.wixsite.com/jamila-agriculture-d

3.16.2 Plagiarism

The plagiarism detected is 3% by : https ://www.grammarly.com/plagiarism-checker.

3.17 Conclusion

Machine learning models have demonstrated substantial breakthroughs and advantages in the field of agriculture, particularly for crop recommendation, weed and insect identification, and fertilizer optimization. These models have proven to have the ability to increase crop output, decrease losses, and improve agricultural practices.

By utilizing historical data, real-time monitoring, and advanced algorithms, these models provide farmers with valuable insights and decision-support tools. Early disease prediction helps in timely interventions, reducing crop losses and the need for excessive pesticide usage. Crop recommendation models assist farmers in selecting the most suitable crops for specific regions, maximizing yield and profitability. Weed and pest detection models enable targeted treatments, minimizing chemical usage and protecting crops. Fertilizer optimization models optimize nutrient management, enhancing crop growth and minimizing environmental impact.

Future Work

Despite the remarkable progress made in implementing these models, there is still room for further development and improvement. Some potential areas for future work include :

Enhanced Accuracy : Continual refinement and improvement of machine learning algorithms to enhance the accuracy of disease prediction, weed detection, and pest detection models. Incorporating more diverse and extensive datasets can lead to more precise predictions.

Integration of IoT and Sensor Technologies : Integration of Internet of Things (IoT) devices and sensor technologies can provide real-time data on various parameters such as soil moisture, temperature, and humidity. Incorporating this data into the models can enhance their performance and enable more accurate recommendations.

Collaborative Data Sharing : Encouraging collaboration and data sharing among researchers, farmers, and agricultural organizations to build more robust and comprehensive datasets. Open data initiatives can contribute to the development of more accurate and reliable models.

In conclusion, the implementation of machine learning models in plant disease prediction, crop recommendation, weed detection, pest detection, and fertilizer optimization has shown promising results. Continual research and development in these areas, along with the integration of emerging technologies, will pave the way for more advanced and effective agricultural practices, contributing to sustainable food production and ensuring global food security.

3.18 Conclusion :

In conclusion, the development of an IoT-based monitoring system for precision agriculture, with a focus on epidemic disease control, offers significant benefits. By providing real-time environmental monitoring and early prediction capabilities, this system allows farmers to maintain optimal crop conditions and take preventive measures against disease outbreaks. The integration of a wireless sensor network and a database enables the collection and storage of valuable environmental and soil data. This data can be accessed and monitored remotely using internet-enabled devices, providing convenience and flexibility to farmers. Additionally, the incorporation of artificial intelligence and prediction algorithms enhances the system's decision-making abilities, emulating human expertise in disease identification and issuing timely warning messages to farmers.

Furthermore, the IoT-based monitoring system can contribute to sustainable agriculture practices by reducing the reliance on pesticides and fungicides. By detecting disease outbreaks at an early stage, farmers can implement targeted interventions and optimize the use of chemical treatments. This not only minimizes environmental damage and reduces the risk of pesticide resistance but also improves overall crop health and productivity.

Moreover, the system's potential goes beyond disease control. It can be expanded to support other aspects of precision agriculture, such as soil moisture monitoring, irrigation management, and crop selection based on soil and climatic criteria. The availability of comprehensive data and advanced analytics empowers farmers to make informed decisions, optimize resource utilization, and enhance overall agricultural productivity.

It is important to note that the successful implementation and adoption of this IoT-based monitoring system require collaboration between farmers, technology providers, and agricultural experts. Training and support programs should be established to ensure farmers can effectively utilize the system and interpret the provided information. Additionally, ongoing research and development efforts are crucial to continuously improve the system's accuracy, expand its functionalities, and adapt it to diverse agricultural contexts.

Overall, the IoT-based monitoring system for precision agriculture presents a promising approach to address the challenges of disease control and optimize agricultural practices. By leveraging advanced technologies, data analytics, and expert systems, it offers a pathway towards sustainable and efficient farming, contributing to food security and environmental preservation.

Bibliographie

[1] Hammoud Dib Abd al-Razzaq,"Diagnosis of plant diseases", 2011.

[2] Abd al-Rahman al-Wakil,"Fundamentals of Plant Diseases by Muhammad ".

[3] Větrovský T, Kohout P, Kopecký M, Machac A, Man M, Bahnmann BD, Brabcová V, Choi J,Meszárošová L, Human ZR, et al. 2019. A meta-analysis of global fungal distribution reveals climatedriven patterns. Nature Communications 10 : 5142.

[4] Delgado-Baquerizo M, Guerra CA, Cano-Díaz C, Egidi E, Wang J-T, Eisenhauer N, Singh BK,Maestre FT. 2020. The proportion of soil-borne pathogens increases with warming at the global scale.Nature Climate Change 10 : 550-554.

[5] STRANGE, R.N. ; SCOTT, P.R. Plant disease : A threat to global food security. Annual Review of Phytopathology, v.43, p.83- 116, 2005.

[6] AGRIOS, G.N. Plant pathology. 5 ed. London : Elsevier, 2004. 922p

[7] Ghini, R., Hamada, E., & Bettiol, W. (2008). Climate change and plant diseases. Scientia Agricola, 65, 98-107.

[8] Ayaz, M., Ammad-Uddin, M., Sharif, Z., Mansour, A., & Aggoune, E. H. M. (2019). Internet-of-Things (IoT)-based smart agriculture : Toward making the fields talk. IEEE access, 7, 129551-129583.

[9] Richard, B., Qi, A., & Fitt, B. D. (2022). Control of crop diseases through Integrated Crop Management to deliver climate-smart farming systems for low-and high-input crop production. Plant Pathology, 71(1), 187-206.

[10] Lowe, A., Harrison, N., & French, A. P. (2017). Hyperspectral image analysis techniques for the detection and classification of the early onset of plant disease and stress. Plant methods, 13(1), 80.

[11] Deutsch, C.A. ; Tewksbury, J.J. ; Tigchelaar, M. ; Battisti, D.S. ; Merrill, S.C. ; Huey, R.B. ; Naylor, R.L. Increase in crop losses to insect pests in a warming climate. Science 2018, 361, 916–919.

[12] Domingues, T. ; Brandão, T. ; Ferreira, J.C. Machine Learning for Detection and Prediction of Crop Diseases and Pests : A Comprehensive Survey. Agriculture 2022, 12, 1350. https ://doi.org/10.3390/agriculture12091350.

[13] Herbert, D.A. ; Mack, T. ; Reed, R.B. ; Getz, R. Degree-Day Maps for Management of Soybean Insect Pests in Alabama ; Auburn University : Auburn, AL, USA, 1988.

[14] Skawsang, S. ; Nagai, M. ; K. Tripathi, N. ; Soni, P. Predicting rice pest population occurrence with satellite-derived crop phenology, ground meteorological observation, and machine learning : A case study for the Central Plain of Thailand. Appl. Sci. 2019, 9, 4846.

[15] Richard, B., Qi, A., & Fitt, B. D. (2022). Control of crop diseases through Integrated Crop Management to deliver climate-smart farming systems for low-and high-input crop production. Plant Pathology, 71(1), 187-206.

[16] PlantVillage Tomato | Diseases and Pests, Description, Uses, Propagation. Available online : https ://plantvillage.psu.edu/topics/tomato/infos (accessed on 13 April 2023).

[17] Liu, Z., Bashir, R. N., Iqbal, S., Shahid, M. M. A., Tausif, M., & Umer, Q. (2022). Internet of Things (IoT) and machine learning model of plant disease prediction–blister blight for tea plant. IEEE Access, 10, 44934-44944.

[18] Dake, W. ; Chengwei, M. The support vector machine (SVM) based near-infrared spectrum recognition of leaves infected by the leafminers. In Proceedings of the

First International Conference on Innovative Computing, Information and Control-Volume I (ICICIC'06), Beijing, China, 30 August–1 September 2006 ; Volume 3, pp. 448–451.

[19] Yeser Abd al-hakim, "The useful acronym for fertilizers and fertilization “.

[20] Šulc, M., and Matas, J. : 'Fine-grained recognition of plants from images', Plant methods, 2017, 13, (1), pp. 115.

[21] How to predict plant diseases.html view 02/16/2023 15 :45 PM

[22] Publication _ USDA ARS.html view 02/19/2023 15 :32 PM

[23] Fundamentals of Plant Pathology _ MEASUREMENT OF PLANT DISEASES.html view 02/19/2023 15 :35 PM

[24] Chiang, K. S., Liu, H. I., & Bock, C. H. (2017). A discussion on disease severity index values. Part I : warning on inherent errors and suggestions to maximise accuracy. Annals of Applied Biology, 171(2), 139-154.

[25] The Arabic Encyclopedia _ Plant Diseases (Mathematical Modeling Trends in).html view 02/19/2023 14 :33 PM

[26] G.Manoj Reddy, Rao Suraj Rao et al, "Smart Farming Crop Yield Prediction using Machine Learning", p45.

[27] P. Wang, B. A. Hafshejani, and D. Wang, "An improved multilayer perceptron approach for detecting sugarcane yield production in IoT based smart agriculture," Microprocessors and Microsystems, vol. 82, article 103822, 2021.

[28] J. Arshad, M. Aziz, A. A. Al-Huqail et al., "Implementation of a LoRaWAN based smart agriculture decision support system for optimum crop yield," Sustainability, vol. 14, no. 2, p. 827, 2022.

[29] M. Cicioğlu and A. Çalhan, "Smart agriculture with internet of things in cornfields," Computers and Electrical Engineering, vol. 90, article 106982, 2021.

[30] H. N. Saha, R. Roy, M. Chakraborty, and C. Sarkar, “Development of IoT-based smart security and monitoring devices for agriculture,” Agricultural informatics : automation using the IoT and machine learning, pp. 147–169, 2021.

[31] P. S. Chatterjee, N. K. Ray, and S. P. Mohanty, “LiveCare : an IoT-based healthcare framework for livestock in smart agriculture,” IEEE Transactions on Consumer Electronics, vol. 67, no. 4, pp. 257–265, 2021.

[32] Ramakrishnam Raju, S. V. S., Dappuri, B., Ravi Kiran Varma, P., Yachamaneni, M., Verghese, D., & Mishra, M. K. (2022). Design and Implementation of Smart Hydroponics Farming Using IoT-Based AI Controller with Mobile Application System. Journal of Nanomaterials, 2022.

[33] Ramya R, Kiran M, Marimuthu E, Naveen Kumar B, Pavithra G (2020) “Plant Monitoring and Leaf Disease Detection with Classification using Machine Learning MATLAB”. International Journal of Engineering Research & Technology (IJERT) ISSN : 2278-0181 Published by, www.ijert.org RTICCT - 2020 Conference Proceedings.

[34] K. Abhirami, G. Chandra Praba, R. Sugantha Lakshmi, 2021, IOT Based Paddy Crop Disease Identification and Prevention System using Deep Neural Networks and Image Processing, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) ICRADL – 2021 (Volume 09 – Issue 05).

[35] Reddy, Y. H., Sreekaree, J., Ashwak, S. M., Sarma, K. L. P. K., Ali, A., Sha, A. Z., & Varaprasad, R. Plant Leaf Disease Detection using IoT, DL and ML,2023.

[36] S. Ramesh, B. Rajaram, Iot based crop disease identification system using optimization techniques, ARPN J. Eng. Appl. Sci 13 (2018) 1392–1395.

[37] Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. Computers and electronics in agriculture, 145, 311-318.

[38] Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D., 2016. Deep neural networks based recognition of plant diseases by leaf image classification. Com-

putat. Intelligence Neurosci. http ://dx.doi.org/10.1155/2016/3289801. Article ID : 3289801.

[39] Mohanty, S.P., Hughes, D.P., Salathé, M., 2016. Using deep learning for image-based plant disease detection. Front. Plant Sci. 7http ://dx.doi.org/10.3389/fpls.2016.01419. Article : 1419.

[40] McCarthy, J. The Logic and Philosophy of Artificial intelligence, 1988.

[41] Russell, R. (2020). Machine Learning : Step-by-Step Guide To Implement Machine Learning Algorithms with Python. (Knxb).

[42] www.javatpoint.com, view 20/05/2023.

[43] 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.

[44] Hinton GE, Salakhutdinov R. Reducing the dimensionality of data with neural networks. Science. 2006 ;313(5786) :504–7.

[45] Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. Computers and electronics in agriculture, 145, 311-318.

[46] Liu W, Wang Z, Liu X, et al. A survey of deep neural network architectures and their applications. Neurocomputing. 2017 ;234 :11–26.

[47] Liu, J., & Wang, X. (2021). Plant diseases and pests detection based on deep learning : a review. Plant Methods, 17, 1-18.

[48] Bengio Y, Courville A, Vincent P. Representation learning : a review and new perspectives. IEEE Trans Pattern Anal Mach Intell. 2013 ;35(8) :1798–828.

[49] Szegedy, C., Com, S.G., 2015. Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift, vol. 37.

[50] www.guru99.com, view 20/05/2023.

[51] Harte, Emma. (2020). Plant Disease Detection using CNN.10.13140/RG.2.2.36485.99048.

[52] Lu, J. ; Tan, L. ; Jiang, H. Review on Convolutional Neural Network (CNN) Applied to Plant Leaf Disease Classification. Agriculture 2021, 11, 707. https :// doi.org/10.3390/agriculture11080707.

[53] Orchi, H. ; Sadik, M. ; Khaldoun, M. On Using Artificial Intelligence and the Internet of Things for Crop Disease Detection : A Contemporary Survey. Agriculture 2022, 12, 9. https ://doi.org/10.3390/ agriculture12010009.

[54] Janiesch, C. ; Zschech, P. ; Heinrich, K. Machine learning and deep learning. Electron. Mark. 2021, 1–11.

[55] Choudhuri, K.B.R. ; Mangrulkar, R.S. Data Acquisition and Preparation for Artificial Intelligence and Machine Learning Applications. In Design of Intelligent Applications Using Machine Learning and Deep Learning Techniques ; Chapman and Hall/CRC : Boca Raton, FL, USA, 2021 ; pp. 1–11.

[56] Akbar, S., Ahmad, K. T., Abid, M. K., & Aslam, N. (2022). Wheat Disease Detection for Yield Management Using IoT and Deep Learning Techniques.

[57] Garcia-Lamont, F. ; Cervantes, J. ; López, A. ; Rodriguez, L. Segmentation of images by color features : A survey. Neurocomputing 2018, 292, 1–27.

[58] Duarte-Carvajalino, J.M. ; Alzate, D.F. ; Ramirez, A.A. ; Santa-Sepulveda, J.D. ; Fajardo-Rojas, A.E. ; Soto-Suárez, M. Evaluating late blight severity in potato crops using unmanned aerial vehicles and machine learning algorithms. Remote Sens. 2018, 10, 1513.

[59] Xiao, Q. ; Li, W. ; Kai, Y. ; Chen, P. ; Zhang, J. ; Wang, B. Occurrence prediction of pests and diseases in cotton on the basis of weather factors by long short term memory network. BMC Bioinform. 2019, 20, 688.

[60] Kaundal, R. ; Kapoor, A.S. ; Raghava, G.P. Machine learning techniques in disease forecasting : A case study on rice blast prediction. BMC Bioinform. 2006, 7, 485.

[61] Wu, X. ; Zhan, C. ; Lai, Y.K. ; Cheng, M.M. ; Yang, J. Ip102 : A large-scale benchmark dataset for insect pest recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019 ; pp. 8787–8796.

[62] Gutierrez, A. ; Ansuategi, A. ; Susperregi, L. ; Tubío, C. ; Ranki´c, I. ; Lenža, L. A benchmarking of learning strategies for pest detection and identification on tomato plants for autonomous scouting robots using internal databases. J. Sens. 2019, 2019, 5219471.

[63] Mohanty, S.P. ; Hughes, D.P. ; Salathé, M. Using deep learning for image-based plant disease detection. Front. Plant Sci. 2016, 7, 1419.

[64] Kamilaris, A. ; Prenafeta-Boldú, F.X. Deep learning in agriculture : A survey. Comput. Electron. Agric. 2018, 147, 70–90.

[65] Sladojevic, S. ; Arsenovic, M. ; Anderla, A. ; Culibrk, D. ; Stefanovic, D. Deep neural networks based recognition of plant diseases by leaf image classification. Comput. Intell. Neurosci. 2016, 2016, 3289801.

[66] Vijay Singh, A.K. Misra (2016) "Detection of plant leaf diseases using image segmentation and soft computing techniques". Elsevier, Information processing in agriculture 4, issue 1.

[67] Albanese, A. ; d'Acunto, D. ; Brunelli, D. Pest detection for precision agriculture based on iot machine learning. In International Conference on Applications in Electronics Pervading Industry, Environment and Society ; Springer : Berlin/Heidelberg, Germany, 2019 ; pp. 65–72.

[68] Gupta, A., & Nahar, P. (2022). Classification and yield prediction in smart agriculture system using IoT. Journal of Ambient Intelligence and Humanized Computing, 1-10.

[69] Pu Y, Gan Z, Henao R, et al. Variational autoencoder for deep learning of images, labels and captions [EB/OL]. 2016–09–28. arxiv :1609.08976.

[70] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. In : Proceedings of the 2014 conference on advances in neural information processing systems 27. Montreal : Curran Associates, Inc. ; 2014. p. 2672 80.

[71] Kamal KC, Yin Z, Wu M, Wu Z. Depthwise separable convolution architectures for plant disease classification. Comput Electron Agric. 2019 ;165 :104948.

[72] What is Computer Vision_ _ IBM.html, view April 12, 2023, 10 :54 :15 AM.

[73] What is precision agriculture_precision farming_.html, view April 9, 2023, 8 :33 :34 AM.

[74] Thirukkumaran, R., Rajalakshmi, B., Priyedarshni, A., Abhigna, K. V., & Kumar, A. (2022, January). Soil and Crop Health Analysis Using IoT and ML. In 2022 International Conference for Advancement in Technology (ICONAT) (pp. 1-4). IEEE.

[75] Ramakrishnam Raju, S. V. S., Dappuri, B., Ravi Kiran Varma, P., Yachamaneni, M., Verghese, D., & Mishra, M. K. (2022). Design and Implementation of Smart Hydroponics Farming Using IoT-Based AI Controller with Mobile Application System. Journal of Nanomaterials, 2022.

[76] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A vision of IoT : Applications,challenges, and opportunities with china perspective," IEEE Internet Things J., vol. 1, no. 4, pp. 349–359, Aug. 2014.

[77] P. P. Ray, "Internet of Things for smart agriculture : Technologies, practices and future direction," J. Ambient Intell. Smart Environ., vol. 9, no. 4, pp. 395–420, 2017

[78] Farooq, M. S., Riaz, S., Abid, A., Abid, K., & Naeem, M. A. (2019). A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming. Ieee Access, 7, 156237-156271.

[79] The Constrained Application Protocol (CoAP), document RFC 7252, 2014. Accessed : Jun. 1, 2019. [Online]. Available : https ://tools.ietf.org/pdf/rfc7252.pdf

[80] O. Deschambault, A. Gherbi, and C. LØgarØ, "Efficient implementation of the MQTT protocol for embedded systems," J. Inf. Process. Syst., vol. 13, no. 1, pp. 26–39, 2017.

[81] I. Grigorik, "Making the Web faster with HTTP 2.0," Commun. ACM, vol. 56, no. 12, pp. 42–49, 2013.

[82] Y. Xiao, H. H. Chen, B. Sun, R. Wang, and S. Sethi, "MAC security and security overhead analysis in the IEEE 802.15.4 wireless sensor networks," EURASIP J. Wireless Commun. Netw., vol. 2006, no. 2, p. 81, 2006.

[83] What is IoT (Internet of Things) and How Does it Work_ - Definition from TechTarget.com.html, view April 9, 2023, 11 :47 :19 AM.

[84] Lova Raju, K., & Vijayaraghavan, V. (2020). IoT technologies in agricultural environment : A survey. Wireless Personal Communications, 113(4), 2415–2446. https ://doi.org/10.1007/s11277-020-07334-x.

[85] Ojha, T., Misra, S., & Raghuwanshi, N. (2015). Wireless sensor networks for agriculture : The stateof-the-art in practice and future challenges. Computers And Electronics In Agriculture, 118, 66–84. https ://doi.org/10.1016/j.compag.2015.08.011.

[86] Lova Raju, K., & Vijayaraghavan, V. (2022). A self-powered, real-time, NRF24L01 IoT-based cloud-enabled service for smart agriculture decision-making system. Wireless Personal Communications, 124(1), 207-236.

[87] Sadowski, S., & Spachos, P. (2020). Wireless technologies for smart agricultural monitoring using internet of things devices with energy harvesting capabilities. Computers and Electronics in Agriculture, 172, 105338. https ://doi.org/10.1016/j.compag.2020.105338

[88] Feng, X., Yan, F., & Liu, X. (2019). Study of wireless communication technologies on Internet of Things for precision agriculture. Wireless Personal Communications, 108(3), 1785–1802. https ://doi.org/10.1007/s11277-019-06496-7

[89] J. R. Millan-Almaraz, R. de Jesus Romero-Troncoso, R. G. Guevara-Gonzalez, L. M. Contreras-Medina, R. V. Carrillo-Serrano, R. A. Osornio-Rios, C. Duarte-Galvan, M. A. Rios-Alcaraz, and I. Torres-Pacheco, "FPGA-based fused smart sensor for real-time plant-transpiration dynamic estimation," Sensors, vol. 10, no. 9, pp. 8316–8331, 2010.

[90] M. I. Husni, M. K. Hussein, M. S. B. Zainal, A. Hamzah, D. M. Nor, and H. Poad, "Soil moisture monitoring using field programmable gate array," Indonesian J. Elect. Eng. Comput. Sci., vol. 11, no. 1, pp. 169–174, Jul. 2018.

[91] S. C. Murray, "Optical sensors advancing precision in agricultural production," Photon. Spectra, vol. 51, no. 6, p. 48, Jun. 2018.

[92] F. P. Povh and W. de Paula Gusm^{a}o dos Anjos, "Optical sensors applied in agricultural crops," Opt. Sensors-New Develop. Practical Appl., 2014.

[93] F. J. García-Ramos, M. Vidal, A. BonØ, H. Malón, and J. Aguirre, "Analysis of the air flow generated by an air-assisted sprayer equipped with two axial fans using a 3D sonic anemometer," Sensors, vol. 12, no. 6, pp. 7598–7613, 2012.

[94] T. K. Yew, Y. Yusoff, L. K. Sieng, H. C. Lah, H. Majid, and N. Shelida, "An electrochemical sensor ASIC for agriculture applications," in Proc. 37th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO), Opatija, Croatia, May 2014, pp. 85–90.

[95] S. Navulur and M. N. Giri Prasad, "Agricultural management through wireless sensors and Internet of Things," Int. J. Elect. Comput. Eng., vol. 7, no. 6, pp. 3492–3499, 2017.

[96] A. Crabit, F. Colin, J. S. Bailly, H. Ayroles, and F. Garnier, “Soft water level sensors for characterizing the hydrological behaviour of agricultural catchments,” Sensors, vol. 11, no. 5, pp. 4656–4673, 2011.

[97] T. Mark and T. Griffin, “Defining the barriers to telematics for precision agriculture : Connectivity supply and demand,” presented at the SAEA Annu. Meeting, Austin, TX, USA, 2016.

[98] What is Cloud Computing_ Pros and Cons of Different Types of Services.html, view April 9, 2023, 11 :16 :05 AM.

[99] K. Bodake, R. Ghate, H. Doshi, P. Jadhav, and B. Tarle, “Soil based fertilizer recommendation system using Internet of Things,” J. Eng. Sci., vol. 1, no. 1, pp. 13–19, 2018

[100] S. Zhang, X. Chen, and S. Wang, “Research on the monitoring system of wheat diseases, pests and weeds based on IOT,” in Proc. 9th Int. Conf. Comput. Sci. Educ., Aug. 2014, pp. 981–985.

[101] Rumpf, T., Mahlein, A. K., Steiner, U., Oerke, E. C., Dehne, H. W., & Plümer, L. (2010). Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance. Computers and electronics in agriculture, 74(1), 91-99.

[102] He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep Residual Learning for Image Recognition, Arxiv.Org, vol. 7, no. 3, pp. 171–180.

[103] Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. Int. Conf. Learn. Represent. 1–14.

[104] Gupta, B., Madan, G., & Md, A. Q. (2022). A smart agriculture framework for IoT based plant decay detection using smart croft algorithm. Materials Today : Proceedings, 62, 4758-4763.

[105] Podder, A. K., Al Bukhari, A., Islam, S., Mia, S., Mohammed, M. A., Kumar, N. M., Cengiz, K., & Abdulkareem, K. H. (2021). IoT based smart agrotech system for verification of Urban farming parameters. Microprocessors and Microsystems, 82, 104025. https ://doi.org/10.1016/j.micpro.2021.104025

[106] Bakthavatchalam, K. ; Karthik, B. ; Thiruvengadam, V. ; Muthal, S. ; Jose, D. ; Kotecha, K. ; Varadarajan, V. IoT Framework for Measurement and Precision Agriculture : Predicting the Crop Using Machine Learning Algorithms. Technologies 2022, 10, 13. https :// doi.org/10.3390/technologies 10010013

[107] H. Kiiski, H. Dittmar, M. Drach, R. Vosskamp, M. E. Trenkel, R. Gutser, and G. Steffens, “Fertilizers, 2. types,” in Ullmann’s Encyclopedia of Industrial Chemistry. 2009.

[108] Mohanty, S.P., Hughes, D.P., Salathé, M., 2016. Using deep learning for image-based plant disease detection. Front. Plant Sci. 7http ://dx.doi.org/10.3389/fpls.2016. 01419. Article : 1419.

[109] Lova Raju, K., & Vijayaraghavan, V. (2022). A self-powered, real-time, NRF24L01 IoT-based cloud-enabled service for smart agriculture decision-making system. Wireless Personal Communications, 124(1), 207-236.

[110] Recurrent Neural Networks for Prediction : Learning Algorithms, Architectures and Stability|Wiley, Wiley.com. Available online : https ://www.wiley.com/engb/Recurrent+Neural+Networks+for+Prediction%3A+Learning+Alg p-9780471495178 (accessed on 13 April 2023).

[111] Plant Disease Detection using Internet of Thing (IoT)2020 Muhammad Amir Nawaz

[112] IOT Based Solution for Leaf Disease Prediction.

[113] DHT11 & DHT22 Sensor Temperature and Humidity Tutorial.html, view April 9, 2023, 8 :47 :25 AM.

[114] Chandan, J., Latha, D., Manisha, R., & Kishore, G. (2022). Monitoring plant health and detection of plant disease using iot and ml. Easy Chair, Tech. Rep.

[115] https ://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/, view 13 April 2023.

[116] Night camera Raspberry pi.mp4, youtube, view March 13, 2023, 9 :38 :22 AM.

[117] Arduino versus Raspberry Pi.mp4 youtube, view March 13, 2023, 9 :31 :23 AM.

[118] https ://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset

[119] https ://www.kaggle.com/datasets/simranvolunesia/pest-dataset, view 20/05/2023

[120] Choudhary, M., Sartandel, R., Arun, A., & Ladge, L. Crop Recommendation System and Plant Disease Classification using Machine Learning for Precision Agriculture. doi, 10, 978 81.

Annexe A : Abbreviations and notation

The following abbreviations are used in this manuscript :

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
GLCM	Grey Level Co-occurrence Matrix
HoG	Histogram of oriented Gradient
DRL	Deep Reinforcement Learning
ResNet	Residual Network
KNN	K-Nearest Neighbor
TL	Transfer Learning
SVM	Support Vector Machine
ML	Machine Learning
DL	Deep Learning
IoT	The Internet of Things
WSN	Wireless sensor network
WCT	wireless communication technologies
MLP	Multi-Layer Perceptron
DSI	Disease Severity Index
PDI	Percent Disease Index