PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research
University of Mohamed Khider – BISKRA
Faculty of Exact Sciences, Natural and Life Sciences
Computer Science Department

N° d'ordre : IA_StartUp 08 /M2/2023

Master Thesis

Presented to obtain the academic master's degree in

Computer science

Option : **Artificial Intelligence (AI)**

Title
Pathology Vegetal Diagnosis based on DCNN technique

---

by :

BARKAT SELSABIL

KERBOUB SARA

Defended on --/06/2023 before a jury composed of:

BOUREKKACHE Samir	MCA	President
SLATNIA Sihem	Professor	Supervisor
TORKI Fatima Zohra	MAA	Examiner

Academic year 2022-2023

Acknowledgements

First and foremost, we would like to express our sincerest gratitude to the Almighty "ALLAH" for bestowing upon us faith, health, strength, courage, patience, and perseverance, and for granting us the resources to complete this humble endeavor. We extend our heartfelt thanks to "ALLAH" for illuminating our path towards success.

We would also like to convey our appreciation to Prof. SLATNIA Sihem, our esteemed supervisor, for her willingness to guide us and for her unwavering patience and invaluable advice. With dedication and genuine interest, she steered this project with enthusiasm, generously sharing her scientific expertise. We extend our deepest gratitude to her, along with our profound admiration for the attention she devoted to this dissertation, her constant encouragement, trust, unfailing availability throughout its completion, and her kindness towards us.

Once again, we extend our warmest thanks to our professors KAZAR Okba and AYAD Soheyb for their timely assistance, invaluable advice, and humble demeanor. We are truly grateful for their contribution to our journey, and we deeply appreciate their unwavering support.

Furthermore, we would like to express our gratitude to each member of the jury, for graciously accepting the responsibility of evaluating our modest work. May they find within these words our sincere appreciation and utmost respect

We express our deepest thanks and heartfelt gratitude to the institution ITDAS which has generously provided us with comprehensive knowledge and information regarding the vast field of agriculture and plants. The institution's unwavering commitment to sharing invaluable insights about plant diseases, along with theoretical and practical instructions, has been instrumental in our professional growth.

We extend our appreciation to all the dedicated engineers and workers within the institution who have selflessly imparted their expertise and guided us throughout our learning journey. Their unwavering support, warm reception, and encouragement have been crucial in propelling us toward success and fostering a positive environment for personal and professional development.

We would like to express our heartfelt thanks to all our friends and colleagues who have provided us with support throughout our research journey. Their assistance, both direct and indirect, has been invaluable in helping us successfully complete our research project.

Dedication

This project is wholeheartedly dedicated to our beloved family.

Our fathers: Ali, Abdelkader.

Our mothers: MERDAS Hayat, TORKI Salima

For their encouragement, their dedication, their sacrifices for our happiness and our success, their love, and their wisdom which allowed us to reach this rank.
No dedication can express my great admiration, consideration, and sincere affection for yous. No matter what we do, we could never reward yous for the great sacrifices yous have made and continue to make for us. May "Allah" protect yous and grant yous a long life full of health and happiness.

Our sisters: Manal, Faten, Hadjer, Raouia

Our Brothers: Zakaria, Yahia, Mohamed Hamza

For their constant presence, their sympathy, their support, and their encouragement they gave me. My pride to whom I wish success.

We extend our heartfelt dedication to the Baghoura family, who warmly welcomed us onto their farm to gather essential and valuable information for our research. We express our sincere gratitude for their gracious hospitality, kindness, and support. Their contribution has been instrumental in ensuring the ongoing progress and success of our work. We deeply appreciate their generosity and assistance.

Abstract

The spread of plant diseases can lead to significant losses in various aspects, including economic and material losses. However, the most severe losses are those that affect human health due to the negative effects of consuming diseased or nutrient-deficient plants. Since tomato cultivation is a crucial agricultural activity in Biskra, Algeria, as it constitutes a large part of the country's agriculture and is cultivated in several regions, we targeted this plant for our study. Tomatoes are susceptible to many diseases, particularly in greenhouse environments where there are optimal conditions for bacteria and insect growth. Despite the widespread interest in tomato cultivation, diseases are still a major concern, particularly in greenhouse environments.

We searched for the types of diseases prevalent in Algeria, specifically in the state of Biskra. It should be noted that these diseases can affect all parts of the plant, including the stem, product, roots and leaves, with certain symptoms indicating their presence. In this study, we focus on diseases that specifically affect leaves, leaving clear symptoms on them.

In our work, we presented an effective method for detecting tomato diseases by making use of deep learning techniques where we depend on the symptoms in both the product and the leaves for the diagnosis. Our approach involved analyzing images from diverse datasets, including those obtained from greenhouses in the city of Biskra. We used convolutional neural network techniques to achieve accurate disease detection and classification.

Keywords — Tomato diseases, Plant Disease, Convolution Neural Network (CNN), Deep Learning, Agriculture, PlantVillage

Résumé

La propagation des maladies des plantes peut entraîner des pertes importantes sous divers aspects, y compris des pertes économiques et matérielles. Cependant, les pertes les plus graves sont celles qui affectent la santé humaine en raison des effets négatifs de la consommation de plantes malades ou carencées en nutriments. La culture de la tomate étant une activité agricole cruciale à Biskra, en Algérie, car elle constitue une grande partie de l'agriculture du pays et est cultivée dans plusieurs régions, nous avons ciblé cette plante pour notre étude. Les tomates sont sensibles à de nombreuses maladies, en particulier dans les serres où les conditions sont optimales pour la croissance des bactéries et des insectes. Malgré l'intérêt généralisé pour la culture de la tomate, les maladies restent une préoccupation majeure, en particulier dans les environnements de serre.

Nous avons recherché les types de maladies prévalant en Algérie, plus précisément dans l'état de Biskra. Il convient de noter que ces maladies peuvent affecter toutes les parties de la plante, y compris la tige, les produits, les racines et les feuilles, certains symptômes indiquant leur présence. Dans cette étude, nous nous concentrons sur les maladies qui affectent spécifiquement les feuilles, laissant des symptômes clairs sur celles-ci.

Dans notre travail, nous avons présenté une méthode efficace pour détecter les maladies de la tomate en utilisant des techniques d'apprentissage en profondeur où nous dépendons des symptômes du produit et des feuilles pour le diagnostic. Notre approche consistait à analyser des images provenant de divers ensembles de données, y compris celles obtenues à partir de serres dans la ville de Biskra. Nous avons utilisé des techniques de réseau neuronal convolutionnel pour obtenir une détection et une classification précises des maladies.

mots clés - Maladies de la tomate, maladies des plantes, réseau de neurones à convolution (CNN), apprentissage en profondeur, agriculture, PlantVillage

Contents

Contents I

List of Figures V

List of Tables VIII

General Introduction 1

1 Machine Learning & Deep Neural Networks 3
- 1.1 Machine Learning & Deep Neural Networks . . . 3
- 1.2 Introduction . . . 3
- 1.3 Machine Learning . . . 3
  - 1.3.1 Definition . . . 3
  - 1.3.2 Types of Machine Learning . . . 4
    - 1.3.2.1 Supervised Learning . . . 4
    - 1.3.2.2 Unsupervised Learning . . . 5
    - 1.3.2.3 Semi-supervised Learning . . . 6
    - 1.3.2.4 Reinforcement Learning . . . 6
- 1.4 Natural neural network . . . 7
- 1.5 Artificial Neural Networks (ANNs) . . . 8
  - 1.5.1 Definition . . . 8
  - 1.5.2 Basic Concepts and Components of ANNs . . . 10
    - 1.5.2.1 Structure of ANNs . . . 10
      - 1.5.2.1.1 Input layer . . . 10
      - 1.5.2.1.2 Hidden layer . . . 10
      - 1.5.2.1.3 Output layer . . . 10
    - 1.5.2.2 Activation functions . . . 11
- 1.6 Deep Learning . . . 13
  - 1.6.1 Definition . . . 13
  - 1.6.2 Deep Learning Architectures . . . 13
    - 1.6.2.1 Recurrent neural networks (RNNs) . . . 13
    - 1.6.2.2 Convolution Neural Network (CNNs) . . . 13
      - 1.6.2.2.1 Definition . . . 13
      - 1.6.2.2.2 Convolution Neural Network Layers . . . 13
      - 1.6.2.2.3 Convolutional Neural Networks for Image Classification . . . 16
      - 1.6.2.2.4 Common CNN Architectures . . . 17
    - 1.6.2.3 Generative Adversarial Networks(GANs) . . . 18
- 1.7 Transfer Learning . . . 18

1.8 The Perceptron . . . 19
1.8.1 Single Layer Perceptron . . . 19
1.8.2 Multilayer perceptron . . . 19
1.9 Conclusion . . . 19

2 Tomato Plant Diseases **21**
2.1 Introduction . . . 21
2.2 Agriculture . . . 21
2.3 Nutrition and Safety of Plant-Based Foods: Impact on Human Health . . . 21
2.4 Tomato Definition . . . 22
2.4.1 Plant forms . . . 24
2.5 Tomato cultivation . . . 24
2.5.1 Cultivation in the open field . . . 24
2.5.2 Cultivation in greenhouse . . . 25
2.6 Economic importance of tomato . . . 26
2.6.1 Worldwide . . . 26
2.6.2 In Algeria . . . 26
2.6.2.1 Tomato area in greenhouses by Wilaya . . . 27
2.6.2.2 Tomato production in greenhouses by Wilaya . . . 27
2.6.2.3 Greenhouse tomato yield by wilaya . . . 28
2.6.3 In Biskra . . . 28
2.7 Tomato Diseases: Overview and Classification . . . 28
2.7.1 Bacterial . . . 29
2.7.2 Insectes . . . 29
2.7.3 Virus . . . 30
2.8 The most common diseases . . . 31
2.8.1 Worldwide . . . 31
2.8.2 In Algeria . . . 32
2.9 Detection Methods for Tomato Diseases . . . 32
2.9.1 Traditionnel . . . 32
2.10 Imaging Techniques and Visual Inspection . . . 33
2.11 Challenges in Tomato Disease Detection . . . 33
2.12 Related Works . . . 33
2.13 Conclusion . . . 34

3 Conception and Implementation **35**
3.1 Introduction . . . 35
3.2 Global Architecture of the system . . . 35
3.3 Detailed Architecture . . . 36
3.4 Dataset . . . 37
3.4.1 Collect Dataset . . . 38
3.4.1.1 Leaf dataset . . . 38
3.4.1.2 Product dataset . . . 38
3.4.2 Preprocessing Dataset . . . 38
3.4.3 Data Labeling . . . 39
3.4.4 Data Spliting . . . 39
3.4.5 Data Resize . . . 39

3.4.6 Data Augmentation . . . 39
3.5 Application architecture . . . 40
3.6 Implementation of the model in IoT system. . . . 40
3.7 CNN learning . . . 41
3.7.1 Training phase of our Model . . . 42
3.7.2 Classification phase . . . 43
3.8 CNN model Evaluation . . . 43
3.8.1 Confusion Matrix . . . 43
3.8.2 Evaluation of the CNN model . . . 44
3.9 Implementation of our deep learning model . . . 44
3.9.1 Hardware configuration . . . 44
3.9.1.1 LENOVO ideapad 320 . . . 44
3.9.1.2 ACER Aspire E5-573 . . . 44
3.9.2 Development environment . . . 45
3.9.3 Packages and APIs . . . 46
3.9.4 Dataset Preparation and Preprocessing . . . 46
3.9.4.1 Data augmentation . . . 46
3.9.4.2 Data resize . . . 47
3.9.4.3 Split Dataset . . . 48
3.9.4.4 Preparing Training set . . . 48
3.9.4.5 Preparing Validation set . . . 49
3.9.4.6 Preparing Testing set . . . 49
3.9.5 Building Our CNN Model . . . 49
3.9.5.1 Import libraries and modules . . . 49
3.9.5.2 Creating CNN model . . . 50
3.9.5.3 Compiling CNN Model . . . 51
3.9.5.4 Model summary . . . 51
3.9.5.5 Training CNN model . . . 53
3.9.6 Testing our CNN model . . . 53
3.10 Extract Frames from Video . . . 53
3.11 Conclusion . . . 55

4 Results and Discussion **57**
4.1 Introduction . . . 57
4.2 Results and discussion . . . 57
4.2.1 leaves Dataset Results . . . 58
4.2.2 Product Dataset Results . . . 60
4.3 Model Complexity and Execution Time Analysis . . . 63
4.4 Visualization of The Application . . . 64
4.4.1 Welcome Page . . . 64
4.4.2 Login Page . . . 65
4.4.3 Sign Up Page . . . 65
4.4.4 Home Page . . . 66
4.4.4.1 Leaf and Tomato Button . . . 67
4.4.4.1.1 Picture Button . . . 68
4.4.4.2 Send Video Button . . . 69
4.4.4.2.1 Diagnostic Process Through Video . . . 69

4.4.4.3 Chat Room Button . . . 71
4.4.5 List Page . . . 72
4.4.5.1 Setting Button . . . 73
4.4.5.1.1 Change Password Button . . . 74
4.4.5.1.2 Change User Name Button. . . . 74
4.4.5.1.3 Rate Us Button. . . . 75
4.4.5.2 Share Button . . . 75
4.4.5.3 About us Button . . . 76
4.4.5.4 Contact us Button . . . 77
4.4.5.5 Logout Button . . . 78
4.4.6 Leaf Detection Diseases Page . . . 78
4.4.7 Tomato Product Detection Diseases Page . . . 80
4.4.8 Diseases Information Page . . . 81
4.5 Conclusion . . . 82

Conclusion General **83**

Bibliography **87**

List of Figures

1.1 Type of Machine Learning[1]. . . . 4
1.2 Supervised Learning[2]. . . . 4
1.3 Overview of supervised learning: Input examples are categorized into a known set of classes[3]. 5
1.4 Unsupervised Learning[2]. . . . 5
1.5 Overview of unsupervised learning, Input samples are grouped into clusters based on the underlying patterns[3]. . . . 6
1.6 Overview of semi-supervised learning. The clusters formed by a large amount of unlabeled data are used to classify a limited amount of labeled data[3]. . . . 6
1.7 Reinforcement Learning[4]. . . . 7
1.8 Biological Neuron[5]. . . . 8
1.9 The Structure of Artificial Neuron[6]. . . . 9
1.10 Biological Neuron and Artificial Neuron[7]. . . . 10
1.11 Sigmoid function[8]. . . . 11
1.12 Rectified linear unit (ReLU)[8]. . . . 12
1.13 Architecture of a Convolutional Neural Network (CNN) [9]. . . . 14
1.14 Convolution Layer[10]. . . . 15
1.15 Max Pooling operation[10]. . . . 15
1.16 Average Pooling operation[11]. . . . 16
1.17 Fully-Connected Neural Network[12]. . . . 16
1.18 The LeNet architecture[13]. . . . 17
1.19 The AlexNet architecture[14]. . . . 17
1.20 The VGGNet architecture[15]. . . . 18
1.21 Multilayer perceptron[16]. . . . 19

2.1 Tomato (Solanum Lycopersicum)[17] 23
2.2 The tomato plant growth stages[18] 24
2.3 Tomato cultivation in the open field[19] 25
2.4 Tomato cultivation in greenhouse[20] 25
2.5 Tomato production percentage of the main producing countries worldwide 2019[21] 26
2.6 Tomato area in greenhouses by wilaya in 2016[22] 27
2.7 Tomato production in greenhouses by wilaya in 2016[22] 27
2.8 Greenhouse tomato yield by wilaya in 2016[22] 28
2.9 Bacterial Spot in Tomato[23] 29
2.10 Whitefly on tomato leaf[18] 30
2.11 Tobacco Mosaic Virus[24] 30

3.1 The General Architecture of Our System. . . . 36
3.2 Detailed Architecture of Our System. . . . 37

3.3 Dataset processing stages. . . . 40
3.4 Application architecture. . . . 41
3.5 IoT system for tomato diseases monitoring. . . . 42
3.6 Our proposed CNN architecture. . . . 42
3.7 Confusion matrix for multi-class classification [25]. . . . 43
3.8 Illustration of Dataset augmentation. . . . 47
3.9 Illustration of Dataset Resize. . . . 47
3.10 Split Dataset code. . . . 48
3.11 Preparing Training Dataset. . . . 48
3.12 Preparing Validation Dataset. . . . 49
3.13 Preparing Testing Dataset. . . . 49
3.14 Necessary libraries for building a CNN model. . . . 50
3.15 CNN model architecture. . . . 51
3.16 Illustration of classification model. . . . 51
3.17 Compiling model. . . . 51
3.18 Model Summary. . . . 52
3.19 Illustration of our model summary. . . . 52
3.20 Training model. . . . 53
3.21 Testing model. . . . 53
3.22 Split Video into Frames 54
3.23 Divide Image 54

4.1 Confusion matrix of leaves Dataset. . . . 58
4.2 Average Accuracy curve. . . . 58
4.3 Average Loss curve. . . . 59
4.4 Confusion matrix of products Dataset. . . . 60
4.5 Average Accuracy curve. . . . 61
4.6 Average Loss curve. . . . 61
4.7 Welcome Page. . . . 64
4.8 Sign in Page. . . . 65
4.9 Sign Up page. . . . 66
4.10 Sign Up page. . . . 66
4.11 Home page. . . . 67
4.12 Import the image to Diagnose it Disease. . . . 67
4.13 Image pick page. . . . 68
4.14 Send Video to Diagnose its Disease.. . . . 69
4.15 Spliting video interface. . . . 70
4.16 Select the sent video. . . . 70
4.17 created files. . . . 71
4.18 Classification phase 71
4.19 Chat Room page. . . . 72
4.20 List Page. . . . 73
4.21 Setting Page. . . . 73
4.22 Change Password page. . . . 74
4.23 Change User Name page. . . . 74
4.24 Rate Us page. . . . 75
4.25 Share Page. . . . 76

4.26 About us Page. 77
4.27 Contact us page. 77
4.28 Logout page. 78
4.29 Leaf Detection Diseases Results. 79
4.30 Product Detection Diseases Results.. 80
4.31 Diseases Information Page. 81

List of Tables

1.1 Comparison of Biological and Artificial Neurons 9

2.1 Tomato composition according to USDA Food Composition Data and Brazilian Table of Food Composition (BTFC) [26] 22
2.2 Worldwide common diseases 31
2.3 The most common diseases in Algeria 32

3.1 CNN layers parameters 41
3.2 Development environment 45
3.3 Packages and APIs 46
3.4 The different layers of DCNN-TD. 50
3.5 CNN Training Parameters. 51

4.1 Classification Report of leaves Dataset. 59
4.2 Classification Report of Product Dataset. 62
4.3 Comparing the Results of Several Dataset Training. 62

General Introduction

Scientific advancements have contributed to helping humans gain a deeper understanding of the environmental conditions that plants require, such as temperature and humidity, and creating mechanisms to provide and control these conditions, such as agricultural greenhouses, for the purpose of producing vegetables out of season, increasing production, and reducing the risk of crop spoilage[27].

Crop diseases are a sensitive issue in agriculture and one of the main factors limiting its sustainability [28]. Tomato cultivation is a significant part of agriculture in Algeria and is practiced in many important climatic regions in the country, including Biskra state [29], famous for producing high-quality tomatoes out of its season by cultivating them in plastic greenhouses. However, the latter provides an ideal environment for the spread of many pests and diseases that threaten crops that are difficult to identify and require early treatment to prevent their spread. Therefore, some farmers lack the experience to identify tomato plant diseases, which may lead to their incorrect estimation of the disease and its improper treatment, ultimately affecting the plant's health, and nutritional value, and reducing production [28].

Field investigations by plant protection experts are a traditional way of warning for tomato diseases and pests are long and slow, hindering timely treatment of diseases and pests. With rapidly growing technology, the involvement of artificial intelligence (AI) in agriculture and vegetation can contribute to sustainable development [30] Therefore, identification methods are needed. Convolutional Neural Networks (CNN) outperform other methods without pre-processing of images. Therefore, they have been used in many fields besides agr iculture[31]. CNN reassembles regular neural networks, but it has a fascinating feature that includes weighted neurons and learned bias. Each neuron takes some input, then performs a dot product, and finally a non-linear product[32].
CNNs operate similarly to traditional neural networks, but they possess unique features such as weighted neurons and learned bias. Each neuron receives input, performs a dot product calculation, and applies a non-linear function to generate an output[33].
Our objective is to develop a software application aimed at assisting individuals engaged in tomato cultivation or studying tomato diseases. This program will empower users to identify different diseases by analyzing images of tomato leaves and other tomato-related products. The software will leverage a comprehensive dataset comprising images of tomato leaves, produits, and tomatoes themselves.

Objective of the Work
This dissertation consists of four chapters, an introduction and a general conclusion. It is organized as follows:

- **General Introduction:** The work commences with a general introduction, providing an overview of the research topic and presenting the fundamental ideas and content explored within.

- **chapter 1:** This chapter covered several important basic concepts in the field of deep learning.
- **chapter 2:** In this chapter, we touched on many topics, including various aspects related to tomato cultivation, and its importance at the global level and in Algeria. We also provided an overview of tomato diseases, their classification, and methods of detection.
- **chapter 3:** In this chapter, the design and implementation process of the system are presented, including an explanation of how the work was conducted, data preparation, and the creation of a comprehensive system structure. The chapter provides a detailed description of the data preparation process, supported by visual illustrations, as well as the code utilized for system development.
- **chapter 4:**In the last chapter,we presented the obtained results, and discussed how the difference in the dataset affects the model.
- **General Conclusion:** Our report encompasses a comprehensive summary and offers insights into future perspectives for this research project.

Chapter 1

Machine Learning & Deep Neural Networks

1.1 Machine Learning & Deep Neural Networks

1.2 Introduction

Deep learning, a branch of artificial neural networks, has emerged as the most widely used computational technique in machine learning due to its ability to learn from large datasets. It has proven to be successful in a wide range of traditional domains, including cybersecurity, natural language processing, bioinformatics, robotics and control, and medical information processing, surpassing established machine learning techniques. The scalability and flexibility of deep learning have made it a popular choice in these fields [34].
One particular class of artificial neural networks called Convolutional Neural Networks (CNN) has gained popularity in computer vision tasks, and is now being applied in various domains, including radiology. CNN can learn spatial hierarchies of features by using multiple building blocks such as convolution layers, pooling layers, and fully connected layers through a backpropagation process[35].
This chapter will provide an introduction to Machine Learning and explore its various types. Additionally, we will delve deeper into Deep Learning and discuss some of its architectures. Finally, we will examine both artificial and biological neurons and their functions.

1.3 Machine Learning

1.3.1 Definition

One of the most renowned AI researchers defines machine learning as a field of research that enables machines to learn without explicit programming. The aim of machine learning is to enhance the efficiency of data processing by machines. Sometimes, the information extracted from data cannot be interpreted by humans, and in such cases, machine learning is applied. Machine learning involves constructing and evaluating algorithms for data analysis. It involves selecting effective features for pattern recognition, classification, and prediction based on models derived from existing data. The demand for machine learning is on the rise due to the availability of vast amounts of data. Mathematicians and programmers use various approaches to address this challenge of handling large datasets [36],[37].

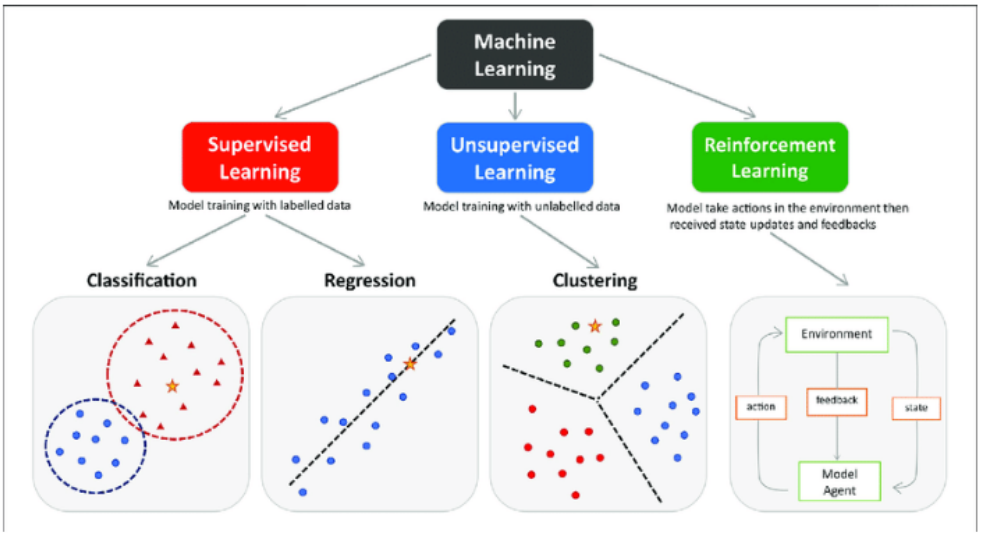


Figure 1.1: Type of Machine Learning[1].

1.3.2 Types of Machine Learning

There are four types of machine learning algorithms:

1.3.2.1 Supervised Learning

Supervised learning is often used in classification problems because the goal is often to teach the computer a classification system that we have created. More generally, learning about classification is appropriate for any problem where the derivation of the classification is useful and the classification is easy to determine. In some cases, it is not even necessary to assign a default classification to each problem if the agent can develop the classification itself. This would be an example of unsupervised learning in the context of classification. Again, number recognition is a common example of classification learning [38].

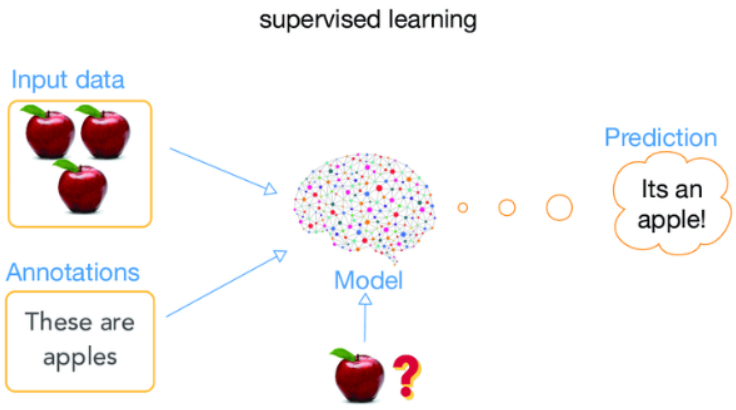


Figure 1.2: Supervised Learning[2].

In supervised learning , regression and classification are two fundamental types of problems. In regression, the output variable is a real or continuous value, such as "price" or "geographical location". On the other hand, classification involves predicting one of a set number of known categories as the output variable, such as "cat" or "dog", "positive" or "negative". Understanding the distinction between these two types of problems is crucial for selecting and applying appropriate machine learning algorithms[3].

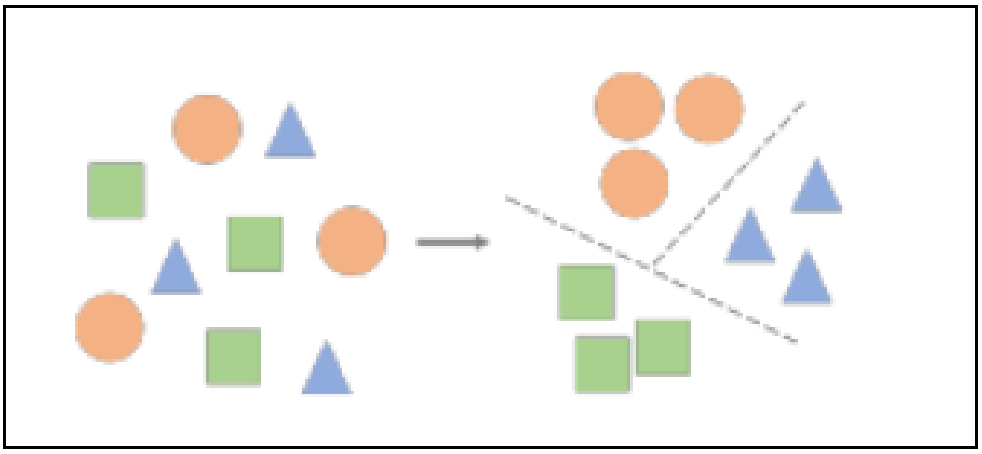


Figure 1.3: Overview of supervised learning: Input examples are categorized into a known set of classes[3].

1.3.2.2 Unsupervised Learning

It's called unsupervised learning because, unlike supervised learning, there are no right answers and no teachers. Algorithms are left alone to discover and represent an interesting data structure. Unsupervised learning algorithms learn various features from data. When new data is entered, it uses previously learned functions to recognize the data class. It is mainly used for grouping and compressing functions[37].

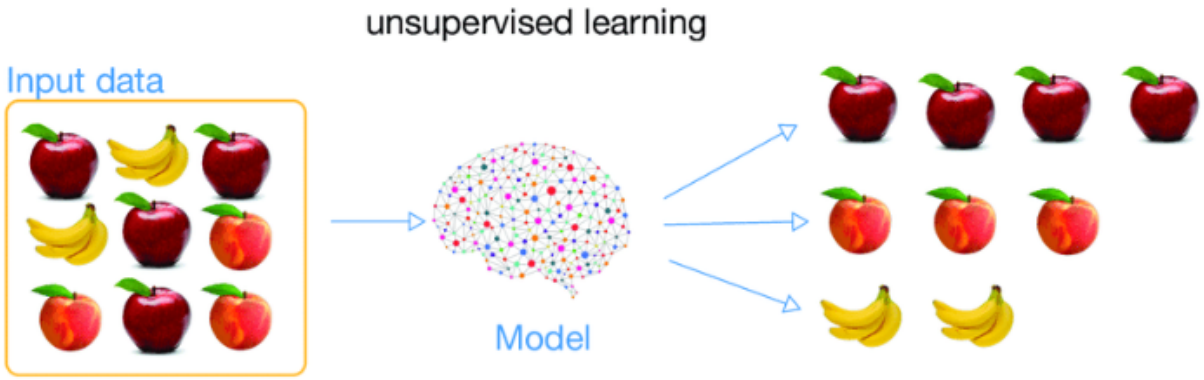


Figure 1.4: Unsupervised Learning[2].

Clustering is a common unsupervised learning technique used in statistical data analysis, machine learning, data mining, pattern recognition, image analysis, and bioinformatics to identify natural groupings or clusters within multidimensional data based on some similarity measure. The process involves grouping similar

objects into different clusters or subsets, according to some defined distance measure. Researchers from various fields actively work on the clustering problem, with different representative clustering methods available[39].

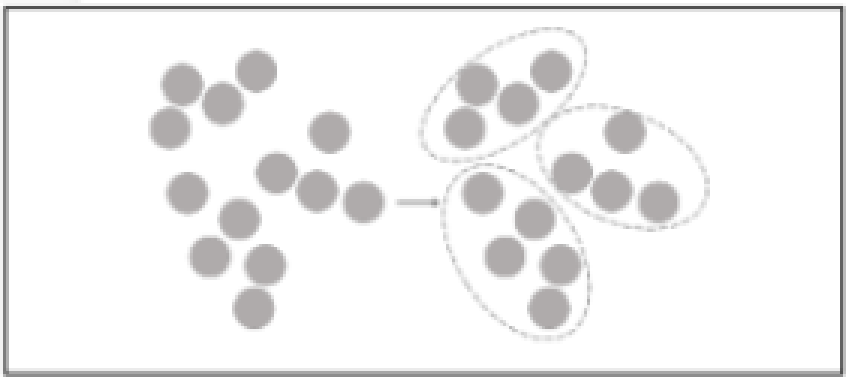


Figure 1.5: Overview of unsupervised learning, Input samples are grouped into clusters based on the underlying patterns[3].

1.3.2.3 Semi-supervised Learning

Semi-supervised learning is a hybridization of the supervised and unsupervised methods because it works on two types of data labeled and unlabeled data. Thus, it falls between "unsupervised" learning and "supervised" learning. In the real world, labeled dates may be rare in various contexts, and unlabeled dates are plentiful where semi-supervised learning is useful. The ultimate goal of a semi-supervised learning model is to produce a better result for the prediction than the one produced using only the tagged data from the model. Some application areas where semi-supervised learning is used, include machine translation, fraud detection, data labeling, and text classification[40].

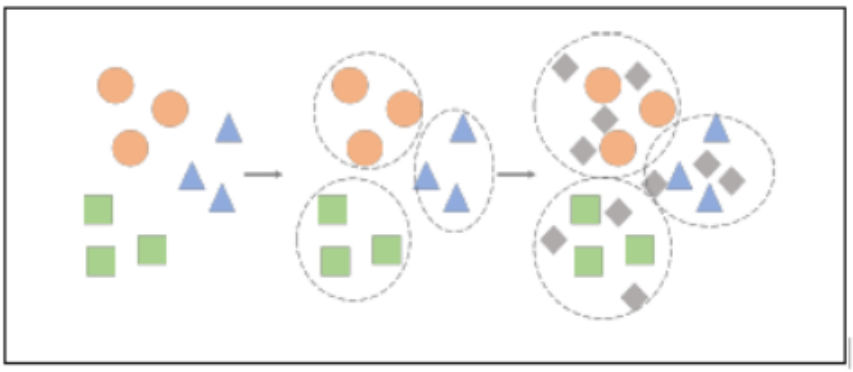


Figure 1.6: Overview of semi-supervised learning. The clusters formed by a large amount of unlabeled data are used to classify a limited amount of labeled data[3].

1.3.2.4 Reinforcement Learning

Reinforcement learning (RL) has a main problem in the learner or the agent included in an environment, where the agent must progressively improve the actions it selects in response to each situation or environ-

mental state. It is critical that, in contrast to supervised learning, the agent does not receive any explicit feedback that directly points to the right actions. Instead, each action triggers an associated reward signal or no reward signal, and the RL problem is to incrementally update the behavior to maximize the cumulative reward over time. Because the agent is not directly told what to do, it must explore alternative actions, collect information about the results they achieve, and so gradually move toward policies of reward maximizing behavior[41].

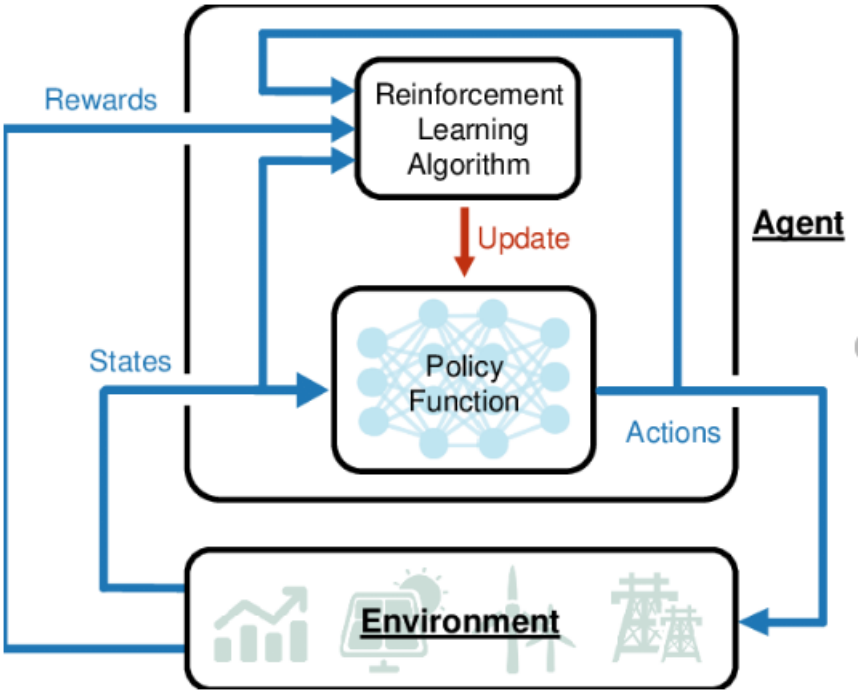


Figure 1.7: Reinforcement Learning[4].

The basic structure of reinforcement learning is depicted in Figure 1.7, where the agent observes the states of the environment and takes actions accordingly. Subsequently, the environment provides a reward based on the actions taken. The primary goal of the agent is to maximize its overall reward throughout the decision period[4].

1.4 Natural neural network

The human brain consists of a lot of interconnected neurons. Each neuron is a cell that performs a simple task, such as responding to an input signal [42]. Network biology can integrate, represent, interpret and model complex biological systems [43]. Natural neurons receive signals through synapses located on the dendrites or neuron membrane. When the received signals are strong enough (above a certain threshold), the neuron fires and sends a signal down the axon. This signal can be sent to another synapse and activate other neurons[44].

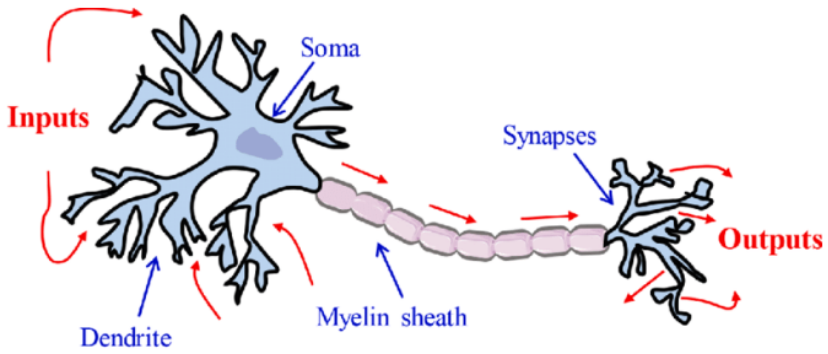


Figure 1.8: Biological Neuron[5].

1.5 Artificial Neural Networks (ANNs)

1.5.1 Definition

Artificial neural networks (ANN), also called neural networks or connectionist models, are mathematical models inspired by the architecture of biological neurons such as the human brain [45]. Artificial neural networks are a set of preprocessors running in parallel and these processors are highly interconnected. Each preprocessor converts the information it receives into an output[46].
In Figure 1.9, a detailed view of a single neuron located in the hidden or output layer is presented. The soma of each neuron is connected to a set of input arcs, labeled as I0 to In. The weights assigned to each arc act as multipliers for any values that are transmitted to the neuron. To compute the output of the neuron, equation (1) is utilized, which involves adding up the values of all the inputs [47].

$$y = f\left(\sum_{j}^{n} W_j I_j + b\right) \tag{1.1}$$

where :
y is the output of the neuron.
f () is the activation function.
Wj is the weight of the inputs.
Ij is the value of the i-th input.
b is the bias term.
In equation (1) an additional term b , has been included called a bias [47].

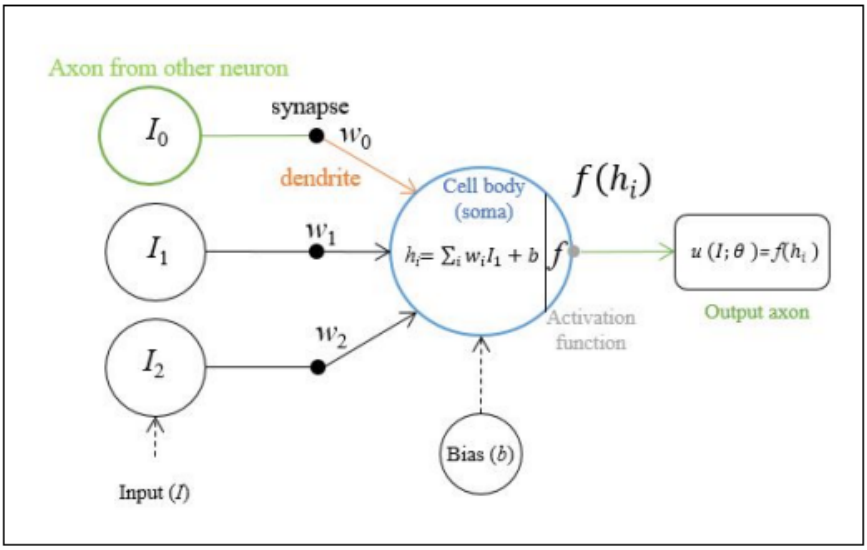


Figure 1.9: The Structure of Artificial Neuron[6].

Figure 1.10, show the differences between Artificial and biological neuron.

Table 1.1: Comparison of Biological and Artificial Neurons

Entity	**Biological Neural Networks**	**Artificial Neural Networks**
Processing Units	Neurons(cell)	Nodes or Units
Inputs	Dendrites	Arcs
Outputs	Axons	Arcs
Interlink age	Synaptic Contact(Synapses)	Node to Node Via Arcs & Interconnection Weighted

ANN comprises three layers: input, hidden, and output. Each layer is composed of a cluster of artificial neurons. The input layer receives input data, which is then processed through the hidden layers, and finally, the output layer produces the desired output. Each layer contains an array of artificial neurons, also called nodes, which process and transmit information. A fully connected neural network contains connections between all neurons in any given layer with those in the previous and next layers. Artificial neurons are mathematical models that emulate various features of biological neurons. Figure 1 illustrates the similarities between the components of biological neurons and their mathematical counterparts. The input layer of the ANN stores a set of input parameters, with each input variable represented by a neuron [7].

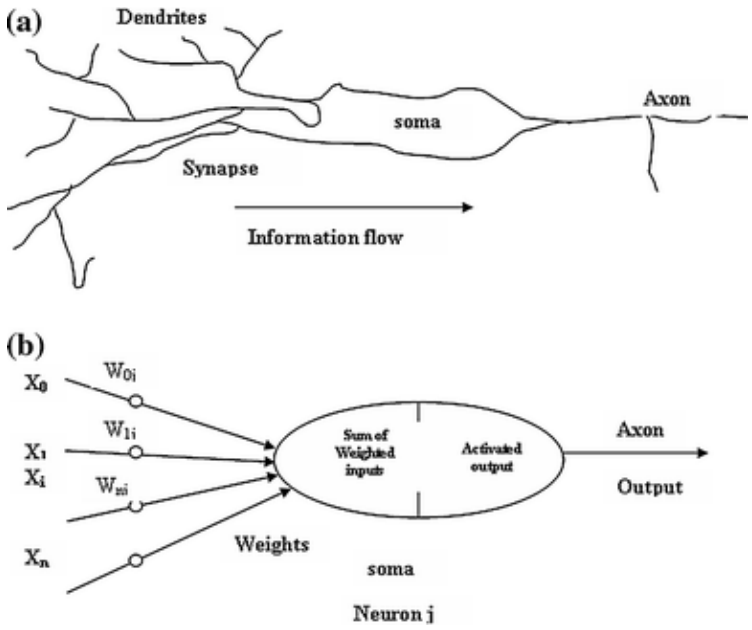


Figure 1.10: Biological Neuron and Artificial Neuron[7].

1.5.2 Basic Concepts and Components of ANNs

In this section we show the basic concepts and components of ANNs:

1.5.2.1 Structure of ANNs

An Artificial Neural Network Architecture consists of three layers.

1.5.2.1.1 Input layer

Data enter the artificial neural network from the outside world through the input layer. This layer is responsible for not only receiving information (data) but also defining signals, characteristics, or measurements. These inputs are usually normalized to the limit values produced by the activation functions. This normalization leads to better numerical accuracy for the mathematical operations performed by the network, after which it moves on to the next layer[48].

1.5.2.1.2 Hidden layer

The hidden layer receives the raw information from input layer and then processes it to extract relevant features and generate output signals for the next layer [49]. The hidden layer is responsible for performing the majority of internal processing within the neural network [48].

1.5.2.1.3 Output layer

Composed of neurons, the output layer is responsible for producing and presenting the final network outputs that result from the processing performed by the neurons in the previous layers of the Artificial Neural Network (ANN) [48].

1.5.2.2 Activation functions

An Activation Function, which is also referred to as a Transfer Function, can be inserted between two neural networks [50]. play a critical role in artificial neural networks by transforming the input signal into an output signal, which is then passed to the next layer in the network. In an artificial neural network, the output of a particular layer is obtained by calculating the sum of products of inputs and their corresponding weights, which is then processed by an activation function. This output is then used as the input for the next layer in the network [8].

1. **Linear Activation Function**
   The activation function is linearly proportional to the input. It can be defined as follows:

$$F(x) = ax \tag{1.2}$$

   The value of variable a can be any constant value selected by the user. Here, the derivative of the function f(x) is not zero but equals the value of the constant used a. There is not much benefit to using a linear function because the neural network will not improve the error due to the same slope value for every iteration. In addition, the network will not be able to identify complex patterns from the data. Therefore, linear functions are ideal when interpretability is required and for simple tasks [8].

2. **Sigmoid function**
   The sigmoid function is a type of function that transforms values into the range of 0 to 1 [8]. It is defined as:

$$F(x) = 1/(1 + e^{(} - x)) \tag{1.3}$$

   A Sigmoid function is a type of real function that is both bounded and differentiable, and is defined for all real input values with a positive derivative at every point. It exhibits a satisfactory level of smoothness and is an appropriate extension of the soft limiting non-linearities used in neural networks. This type of function is characterized by rapidly approaching a fixed, finite upper limit as its argument gets large, and rapidly approaching a fixed, finite lower limit as its argument gets small, approaching these limits asymptotically[51].

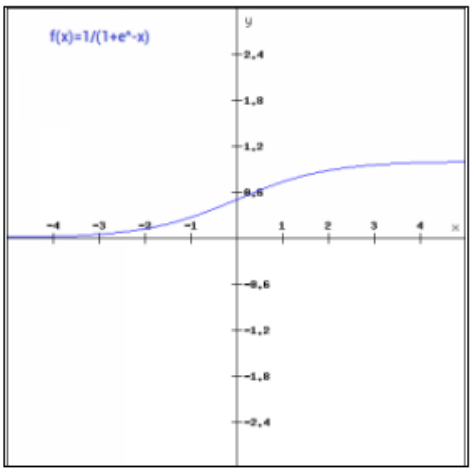


Figure 1.11: Sigmoid function[8].

3. **Rectified linear unit (ReLU)** Rectified linear unit (ReLU) activation has solved the problem of leakage and gradient explosion as a derivative of unity for positive values and many popular networks use it as an activation function for input/hidden layer. The ReLU behaves like an identity function f(x) = x for all positive x and otherwise is said to be :

$$ReLU(x) = max(0, x) \tag{1.4}$$

Even though the differential of ReLU is not a function continuous, but it is defined over the entire input range (1 for positive x and 0 for the opposite). As ReLU continues to be the most common support function for most DL architectures, variants of ReLU are being developed, such as Leaky ReLU Randomized, Leaky ReLU, and Parametrized ReLU [52].

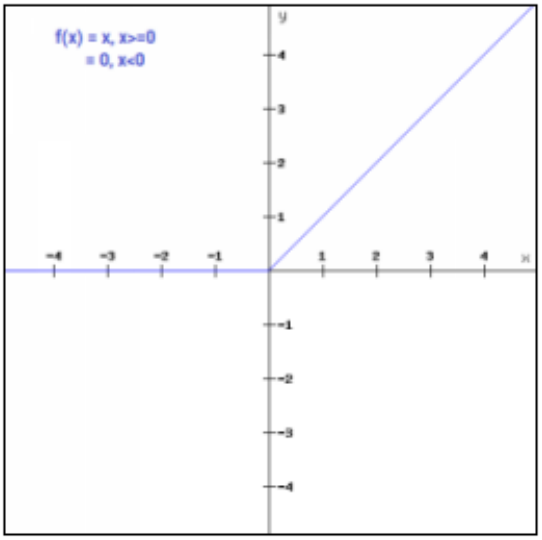


Figure 1.12: Rectified linear unit (ReLU)[8].

4. **Softmax function**
The softmax function is constructed by combining several sigmoid functions, where each sigmoid function produces a probability value between 0 and 1 for the data points belonging to a specific class [8]. The softmax function is frequently utilized as the final layer in deep learning classification tasks, as well as in other operations like attention mechanisms. Despite its widespread usage, the softmax function has been heavily scrutinized, leading to the exploration of alternative approaches. The softmax function, unlike sigmoid functions that are employed for binary classification, is designed for handling multiclass classification problems. For each data point across all classes, the function produces a probability value that corresponds to its respective class[53]. The mathematical expression for the softmax function is given as:

$$\sigma(y_i) = \left(\frac{e^{y_i}}{\sum_j e^{y_j}} \right) j = 1, ..., n[54] \tag{1.5}$$

1.6 Deep Learning

1.6.1 Definition

Deep Learning (DL) is a neural network with a multitude of layers and parameters. Deep learning methods always use neural network architectures. and they are also called deep neural networks and were taken from machine learning[55]. "Deep" is a technical term and refers to the number of ANN layers. There are three types of layers: input layer (receives input data), output layer (produces the result of data processing), and hidden layer (extracts patterns from data). Like surface RNA (single hidden layer), deep RNA has multiple hidden layers that enable it to perform even the most complex tasks. Simply put, DL works great with unstructured data and has higher accuracy than ML, but requires a huge amount of training data and expensive hardware and software[56].

1.6.2 Deep Learning Architectures

The number of architectures and algorithms that are used in deep learning is wide and varied. this section explores deep learning architectures:

1.6.2.1 Recurrent neural networks (RNNs)

Recurrent Neural Networks (RNNs) are a class of neural networks that are well-suited for processing sequential and time-series data. They are designed as an extension to feedforward networks, allowing the processing of variable-length and even infinite-length sequences. RNNs use feedback loops to process data, where the output of a neuron is fed back into the network as input for the next iteration, enabling the network to maintain an internal state or memory of previous inputs. Popular recurrent architectures used in RNNs include long short-term memory (LSTM) and gated recurrent units (GRUs). RNNs have become increasingly popular in recent years, finding applications in various fields, such as natural language processing, speech recognition, and image captioning [57, 58].

1.6.2.2 Convolution Neural Network (CNNs)

1.6.2.2.1 Definition

The creation of Convolutional neural networks (CNN) has been for years ago [59],and have shown big popularity partially because of their success in image classification this day [60]. A convolution neural network is a multi-layer artificial neural network specially made to solve two-dimensional input data problems. witch have different architectures; however, in general, they consist of convolutional and pooling layers, which are grouped into modules. one fully connected layer or more. Modules are placed on top of each other to make a deep model. An image is an input to this kind of network, and this is followed by stages of convolution and pooling. Thereafter, the model feeds one or more fully connected layers for classification. Finally, the last layer outputs the class label [61].

1.6.2.2.2 Convolution Neural Network Layers

CNN analyzes data in only its receptive area In the same way as neurons in the biological vision system Each neuron in the human brain has its own receptive field and is connected to other neurons in order to cover the full visual field[62].

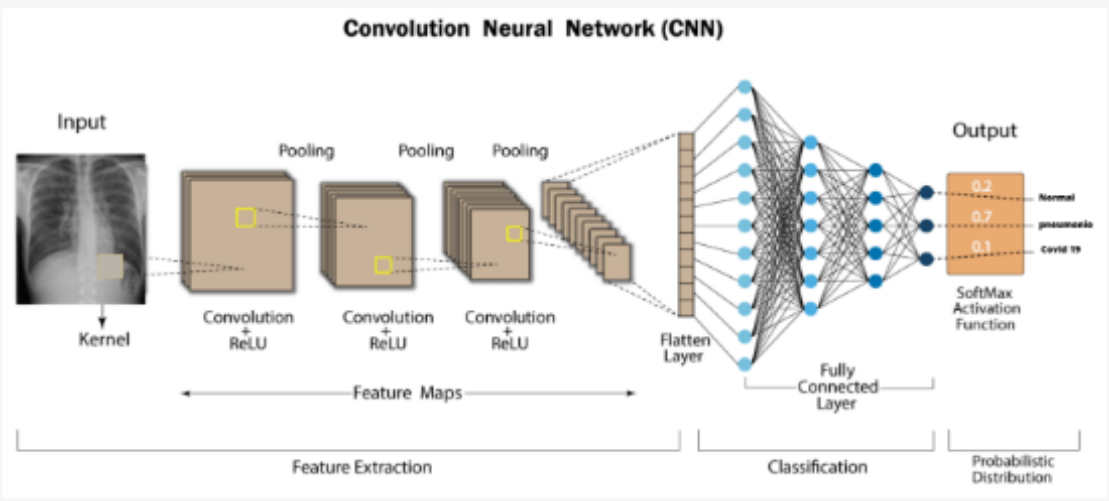


Figure 1.13: Architecture of a Convolutional Neural Network (CNN) [9].

1. **Input Layer**
   A computer image is built of blocks of pixels ranging from 0 to 255, where the pixel value specifies each pixel's brightness and hue [63].

2. **Convolutional Layer**
   The CNN contains a collection of layers that perform feature extraction, using two basic operations:

   2.1 Convolution
   Multiple filters of the convolutional layer slide over the layer for the given input data. Then the sum of the element-by element multiplication of the filters and the input field is calculated as the output of this layer. The total weight is set as an element of the next layer. Each convolution operation is specified by stride, filter size, and zero padding. The stride, which is a positive integer, determines the slip step[64].

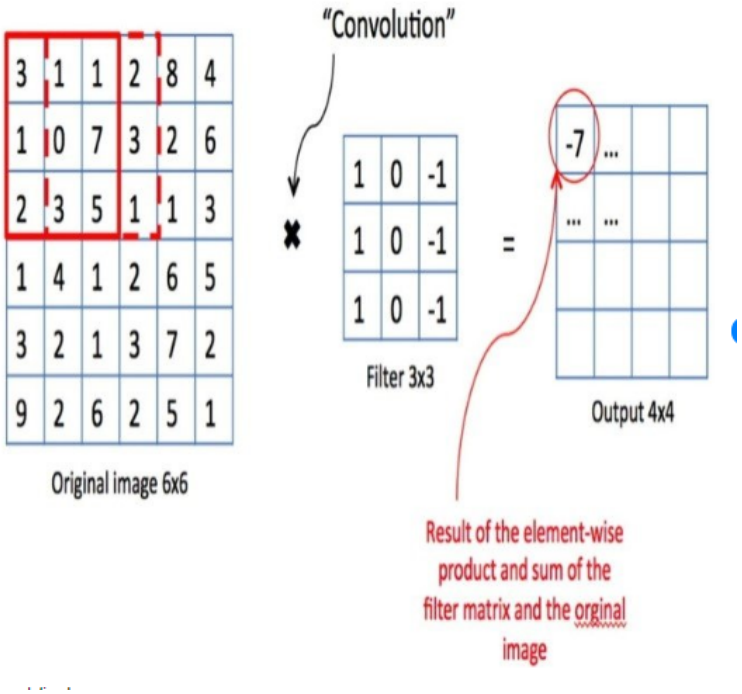


Figure 1.14: Convolution Layer[10].

2.2 Pooling The purpose of pooling is to convert the representation of common features into a more usable one that preserves important information while removing irrelevant details. The use of this layer in CNN aims to achieve invariance to changes in position or lighting conditions, robustness to clutter, and compact representation. In general, the pooling layer summarizes the outputs of neighboring groups of neurons in the same kernel map[65].

2.2.1 Pooling Methods

There are two pooling methods commonly used in CNNs:

- **Max Pooling Method** CNNs are usually using Maximum pooling because of no need to change the parameters. it is a way that helps to optimize the size of the feature map while ensuring translation invariance across the network. which is done by choosing the largest value in the feature map. The Maximum pooling technique identifies the largest element in each joining area[66].

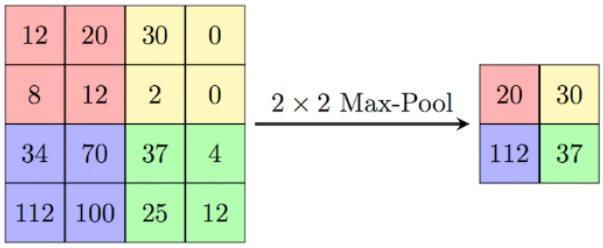


Figure 1.15: Max Pooling operation[10].

- **Average Pooling Method** The average pooling used for the feature extraction, the first convolution-based deep neural network. This layer works by downsampling, taking votes in rectangular areas of the pool, and calculating the averages of each[65].

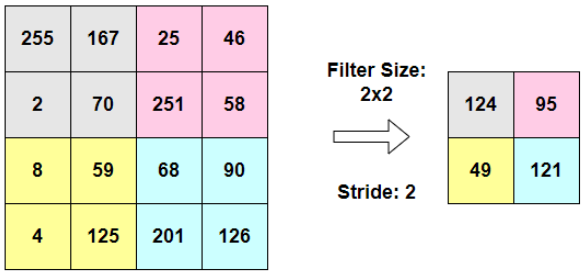


Figure 1.16: Average Pooling operation[11].

3. **Fully Connected Neural Network**
   the final layer in the prosses of the convolutional neural network. Usually, the feature map is flattened before it is passed to the neurons. It is not easy to track the data after the flatting because of the huge number of hidden layers and different weights for the output of each neuron[67].

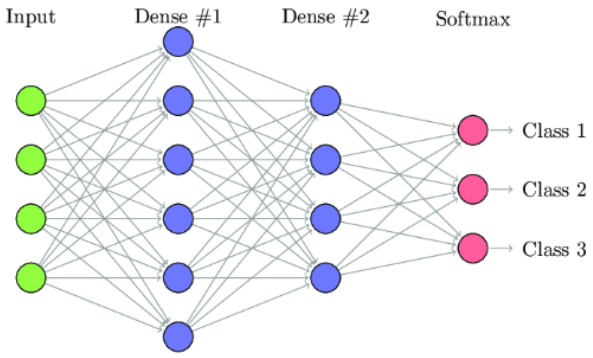


Figure 1.17: Fully-Connected Neural Network[12].

1.6.2.2.3 Convolutional Neural Networks for Image Classification

Computer vision is a study of how to use the simulation of human visual science, and this process is done by collecting images, analyzing them, and processing them to extract their main features and identify the target and the object from the image, as the essence of the classification is to extract the features[68]. In the field of image classification, convolutional neural networks have shown a lot of achievements in extracting features from images. Compared to traditional manual methods of extraction Advantages CNN achieves faster, more reliable and robust results[69]. In order to train a CNN model effectively, it is crucial to have a large and well labeled dataset that corresponds to the specific field of study. For instance, datasets may be available that focus on a group of diseases, where each disease has its own set of files that include images showcasing its distinct symptoms. Additionally, it is important to fine tune the parameters of the model, such as adjusting the tuning grid and input settings (e.g. weights, image size, filter size, number of convolutional layers) to optimize performance[70].

1.6.2.2.4 Common CNN Architectures

Convolutional Neural Networks (CNNs) are a type of neural network used in image recognition, natural language processing, and other areas. Some common CNN architectures include:

1. **LeNet**
   In Convolutional Neural Network the most popular architect that was accustomed to reading zip codes, and digits are LeNet which was developed by Yann LeCun in 1990 and enhanced in 1998[71].

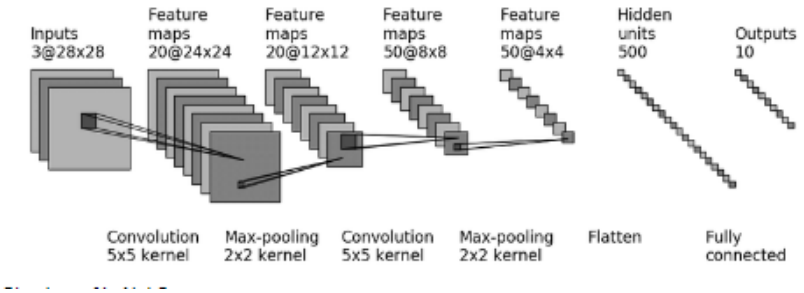


Figure 1.18: The LeNet architecture[13].

2. **AlexNet**
   AlexNet is CNN architecture, which makes the convolutional neural network popular in Computer vision, developed by Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton .it had a similarity to the architecture LeNet. Later, in 2012 AlexNet was presented to the ImageNet ILSVRC challenge and considerably performed better than the second runner up but more profound, more significant architecture[71].

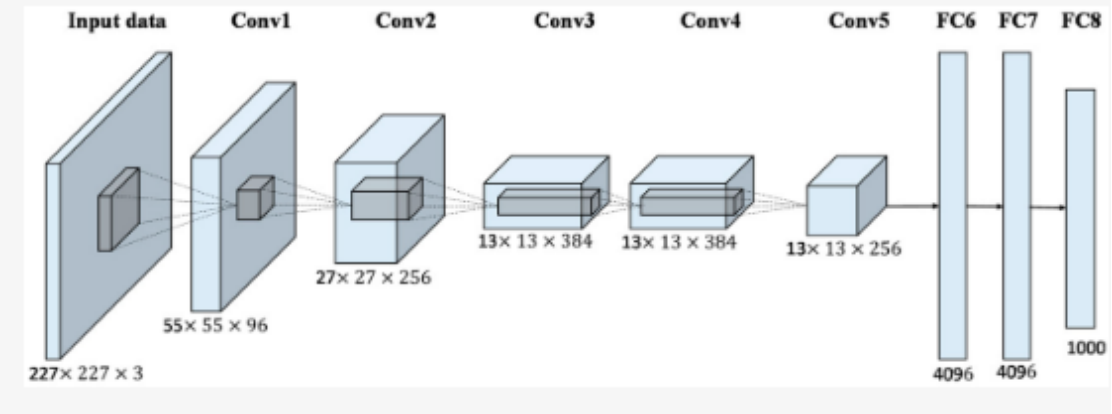


Figure 1.19: The AlexNet architecture[14].

3. **VGGNet**
   Karen Simonyan and Andrew Zisserman are the developers of VGGNet which was created to show that depth is one of the important factors for well given accuracy. the total forme contains 16 CONV/FC layers .3x3 convolutions and 2x2 pooling from the start to the end.. created to show that depth is one of the important factors for well given accuracy. the total form contains 16 CONV/FC layers .3x3 convolutions and 2x2 pooling from the start to the end [71].

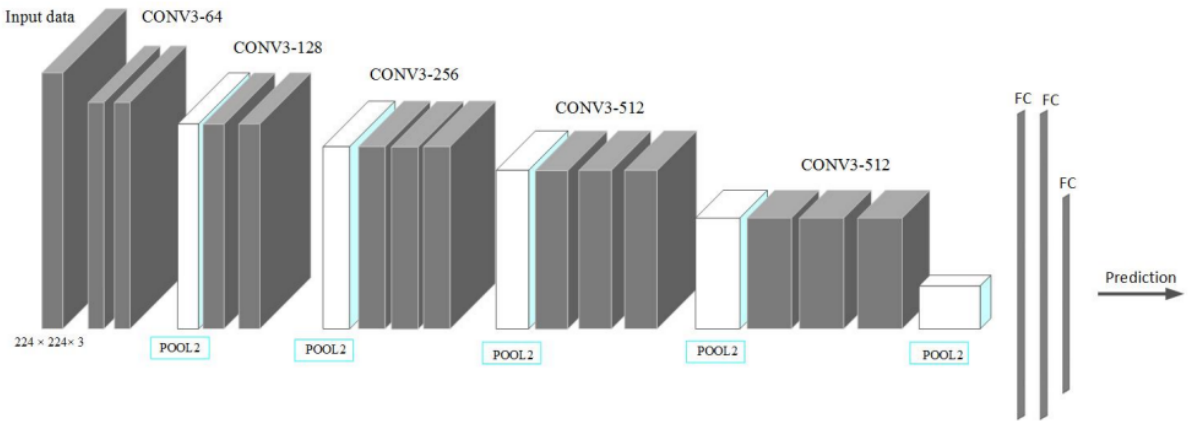


Figure 1.20: The VGGNet architecture[15].

1.6.2.3 Generative Adversarial Networks(GANs)

Adversarial networks are artificial intelligence algorithms developed to find a solution to the problem of generative modeling. It is a modern technique used for learning in both semi supervised and unsupervised mode where the collection of learning examples and learning probability is the intermediate goal of the algorithm. GAN is the most successful generation model as it has been applied to challenging types of tasks, but still shows unique research challenges and opportunities because it is based on game theory while most other approaches to generative modeling are based on optimization. In a previous study, researchers [72] proposed describing this technique through Training a pair of networks to compete with each other proposed describing this technique through Training a pair of networks to compete with each other [73]. GANs are widely considered to be the most successful generative models (especially for their ability to generate photorealistic, high resolution images). It has been implemented for some tasks like creating images from descriptions, optimizing low resolution images, detecting objects, translating images by images, and many other applications [74].

1.7 Transfer Learning

In order to improve a learner's level in a particular domain by transferring information from a related domain, we use transfer learning [75].Inspired by the human mind's ability to transfer knowledge across tasks, we use transfer learning to improve learning performance or reduce the number of labeled examples required in the target domain by utilizing knowledge from the source domain. Transfer learning has been shown to have broad applicability, for example in image classification, sentiment classification , dialogue systems , and urban computing [76]. Transfer learning is a tool in machine learning to find the solution to insufficient training data problems. It is basic to transfer knowledge from the source to the target by minimizing the value of training data and test data not must be independent and identically distributed This will lead to a great positive effect on many domains that are difficult to improve because of insufficient training data[77].

1.8 The Perceptron

The perceptron is one of the most widely used neural networks for approximation,classification, and prediction problems.

1.8.1 Single Layer Perceptron

An artificial neuron, also known as a formal neuron, is a finite algebraic function that is non-linear. Although the complexity of real neurons is difficult to model, artificial neurons are composed of variables referred to as inputs, such as synapses, which are multiplied by parameters known as weights representing the signal intensities, and then computed using a mathematical function that determines the neuron's activation. The output of an artificial neuron is typically calculated by another function, which may be an identity, sometimes based on a specific threshold. Artificial neural networks connect these artificial neurons to process information[44].

1.8.2 Multilayer perceptron

Perceptron Multi-Layer Network is a network model which contains , hidden layers whose function is to perform intermediate processing for better prediction, unlike formal neurons which only have the input and output layers. These architectures make use of functions that are nonlinear such as the hyperbolic tangent function or logistic function to activate neuron outputs. We can educate with these models a network with several hidden layers and neurons. Thus, the more layers there are, the deeper the network and the richer the model becomes. It is Deep Learning [78].

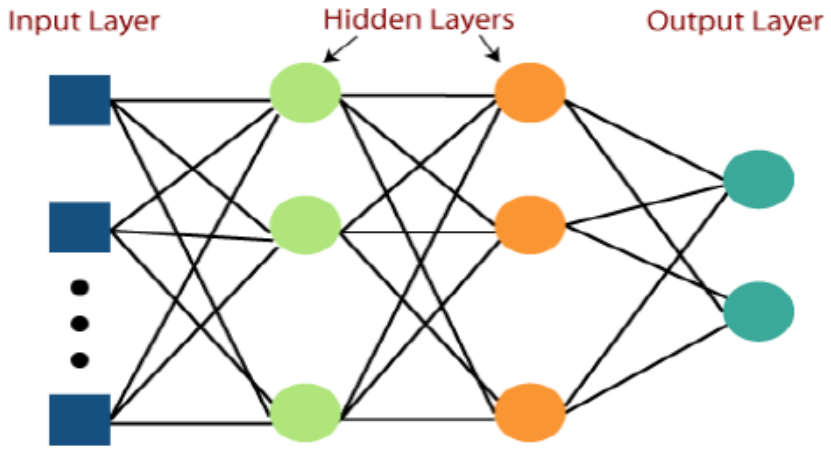


Figure 1.21: Multilayer perceptron[16].

1.9 Conclusion

In this chapter, we provided a comprehensive examination of machine learning and deep learning techniques. We covered fundamental concepts related to model optimization in the field of machine learning, with

a particular focus on Deep Neural Networks classifiers. Additionally, we delved into the intricacies of Convolutional Neural Networks (CNN) and explored their various layers.

Chapter 2

Tomato Plant Diseases

2.1 Introduction

Agriculture plays a crucial role in providing the primary and essential source of income for many countries worldwide. It involves the cultivation of crops and raising animals for various products, including food and fiber. Unfortunately, diseases affecting plants and crops pose a significant threat to their production, leading to economic losses. These diseases manifest in various parts of the plants, but their symptoms are most commonly visible on leaves. In this chapter, we will explore agriculture and plant diseases, with a focus on the tomato plant and their potential impact on human health.

2.2 Agriculture

Agriculture signifies the different ways in which human life is continued with the availability of crop plants and domestic animals by providing food and other products [79].It is a complete system of plant and animal production practices Producing sufficient quantities of high-quality food, preserving resources, and being both environmentally friendly and profitable are what farms should aim for. Instead of using fertilizers, a sustainable farm relies on natural processes and renewable resources sourced from the farm itself. that can benefit : (a) Food and animal self-sufficiency; (b)high-quality environment; (c) Rational and practical use of materials needed for cultivation, biological cycles, and appropriate preparations (d) Contribute to raising the economy in terms of agriculture; and (e)Improving the livelihood of farmers[80].

2.3 Nutrition and Safety of Plant-Based Foods: Impact on Human Health

The relationship between plant health and human health is complex and multifaceted, with numerous factors influencing the overall impact on human health. One important factor is the nutritional content of plants, which can vary greatly depending on factors such as soil quality, weather conditions, and farming practices [81]. For example, studies have shown that organic farming practices can lead to increased levels of beneficial nutrients such as vitamin C, iron, and magnesium in products and vegetables [82]. Additionally, plants that are grown in soil that is rich in nutrients and microorganisms can have higher levels of phytochemicals, which have been linked to a reduced risk of chronic diseases such as cancer and heart disease [83]. On the other hand, the use of pesticides and other chemical fertilizers can have negative impacts on both plant and human health. Exposure to pesticides has been linked to a range of health issues,

including cancer, Parkinson's disease, and reproductive problems [84]. Furthermore, the use of chemical fertilizers can lead to soil degradation and nutrient depletion, which can ultimately impact the nutritional quality of plants [85]. Overall, promoting plant health through sustainable farming practices is essential for ensuring the long-term health and well-being of both plants and humans.

2.4 Tomato Definition

Nowadays, Tomato (Solanum lycopersicum) is an important crop plant grown worldwide, and its production and consumption continue to increase. This popular vegetable is known as a major source of important nutrients including lycopene, beta carotene, flavonoids, vitamin C, as well as hydroxycinnamic acid derivatives. Moreover, this crop has achieved immense popularity, especially in recent years with the discovery of lycopene's antioxidant activities and anti-cancer functions [86].

Table 2.1: Tomato composition according to USDA Food Composition Data and Brazilian Table of Food Composition (BTFC) [26]

Nutrien(Units)	USDA	BTFC
Water (%)	94.50	95.1
Energy (Kcal)	18	15
Energy (Kj)	75	64
Protein (g)	0.88	1.1
Total lipid (fat) (g)	0.20	NA
Vitamin C, total ascorbic acid (mg)	12.7	21.2
Carbohydrate (g)	3.92	3.10
Fiber, total dietary (g)	1.20	1.20
Calcium (mg)	10	7
Iron (mg)	0.27	0.20
Magnesium (mg)	11	11
Phosphorus (mg)	24	20
Potassium (mg)	237	222
Total lipid (fat) (g)	0.20	NA
Vitamin C, total ascorbic acid (mg)	12.7	21.2
Carbohydrate (g)	3.92	3.10
Fiber, total dietary (g)	1.20	1.20
Calcium (mg)	10	7
Iron (mg)	0.27	0.20
Riboflavin (mg)	0.019	Tr

Niacin (mg)	0.594	NA
Manganese (mg)	0.114	0.07
Beta-carotene (µg)	449	NA
Alpha-carotene (µg)	101	NA
Lutein+zeaxanthin (µg)	123	NA
Magnesium (mg)	11	11
Phosphorus (mg)	24	20
Potassium (mg)	237	222
Sodium (mg)	5	1
Lycopene (µg)	2573	NA
Thiamin (mg)	0.037	0.12
Zinc (mg)	0.17	0.10
Copper (mg)	0.059	0.04
Ash (g)	0.50	0.50

Figure 2.1: Tomato (Solanum Lycopersicum)[17]

2.4.1 Plant forms

In cultivated tomatoes, there are two types of tomatoes, indeterminate and determinate, the first being a single vine usually formed to hold a stem with all the side shoots removed and the second ending. ends with a flower cluster cease-fire. lengthen. Definite varieties develop during cold or short seasons and are often earlier than indeterminate varieties. The products in this case ripens almost simultaneously, making this plant suitable for mechanical harvesting. Due to the indeterminate shape, the tree is intended for long-term production, if properly cared for, it will produce products continuously for a long time. The time from planting to commercial maturity for early varieties is 50 to 65 days while for late varieties it is 85 to 95 days [87].

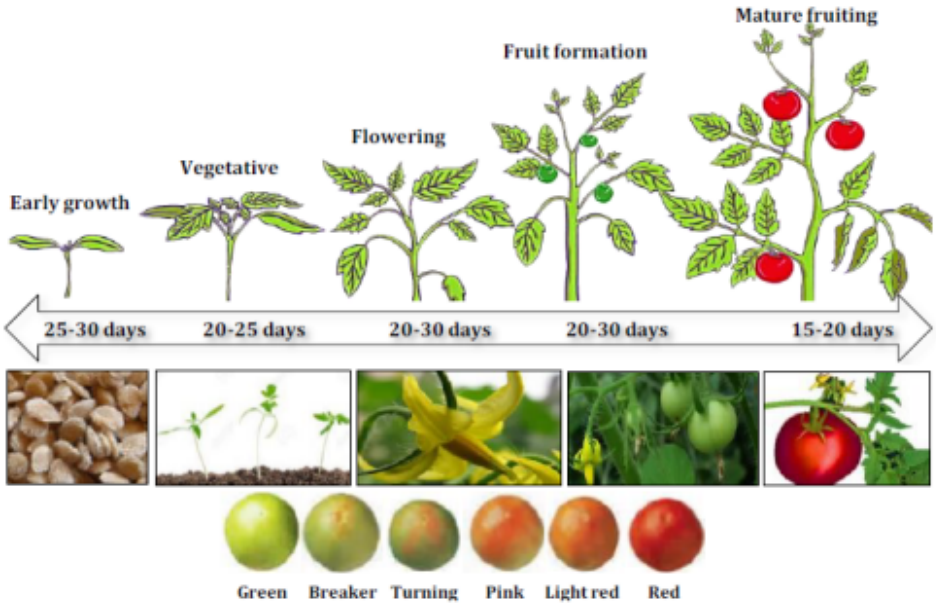


Figure 2.2: The tomato plant growth stages[18]

2.5 Tomato cultivation

Tomato cultivation depends on two methods:

2.5.1 Cultivation in the open field

Open-field farming is generally seen as a "green" activity, one that has much less visual impact. As studies confirmed that the cost of production is lower compared to other methods, and in addition, the price of tomatoes grown in the fields is lower[88].

Figure 2.3: Tomato cultivation in the open field[19]

2.5.2 Cultivation in greenhouse

Tomatoes can be grown in all types of greenhouses, as long as they are tall enough to handle and train the plants vertically. High light transmittance is very important and it varies from 70% to 81% in modern greenhouses. In many countries above 50° north latitude, a Venlo-type greenhouse, consisting of a 1.5-hectare block with a span of 3.2 m, is used[89].In the greenhouse, it is possible to control the environment and those factors that affect the plant's well-being, thereby keeping the tomato plant productive over a long period of time[90]. Despite its high investment cost, multi-chamber greenhouses offer numerous advantages compared to tunnel greenhouses and remain a highly effective tool for protected crop cultivation worldwide, as long as their use is profitable, which can be achieved by producing more at lower costs [91].

Figure 2.4: Tomato cultivation in greenhouse[20]

2.6 Economic importance of tomato

2.6.1 Worldwide

Tomato is considered one of the most economically significant crops globally, with an estimated market value of around USD 182 billion in 2020 [92]. It is the most widely produced vegetable worldwide, with over 180 million tons produced in the same year [93]. Tomatoes are utilized for fresh consumption and processing, such as in ketchup, sauces, juices, and canned tomatoes. The tomato industry provides employment to millions of people worldwide, ranging from growers and harvesters to processors and distributors. Additionally, tomatoes play a vital role in global trade, with many countries exporting and importing large quantities of tomatoes [94].

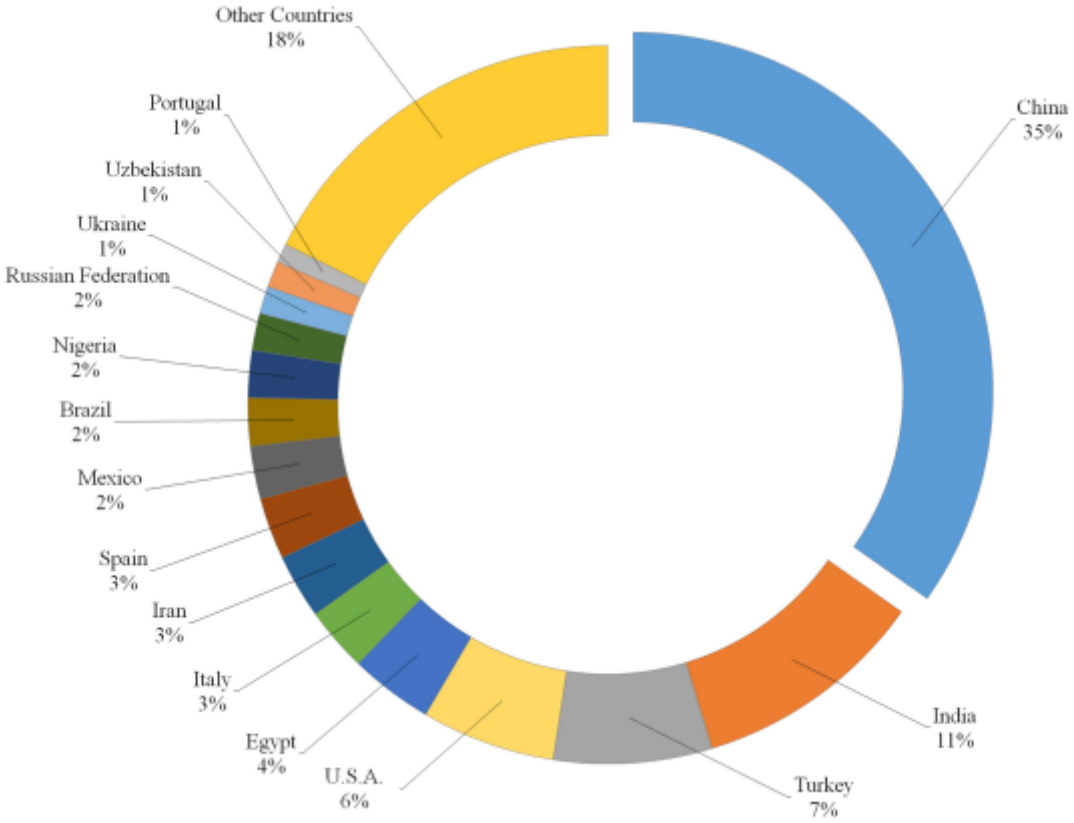


Figure 2.5: Tomato production percentage of the main producing countries worldwide 2019[21]

2.6.2 In Algeria

Tomatoes are one of the most important vegetable crops in Algeria, and they are cultivated in various regions throughout the country [95]. However, Algeria remains a net importer of tomatoes, with an average annual import volume of approximately 350,000 tons [96]. The tomato sector is significant in terms of providing employment opportunities for both urban and rural residents. The Algerian government has made attempts to support tomato production by offering subsidies and investing in irrigation infrastructure [95]. However, the tomato industry in Algeria faces a variety of challenges, including high production costs, low productivity, and competition from less expensive imports [97].

2.6.2.1 Tomato area in greenhouses by Wilaya

In 2016, the Wilaya of Biskra occupies 49% of the national area of tomatoes in greenhouses, the second wilaya is Tipaza which exploits 15% of the national area of tomatoes in greenhouses, the third the wilaya of Chlef (10% of the area) and the fourth Mostaganem (6% of the national area dedicated to the cultivation of tomatoes in greenhouses).

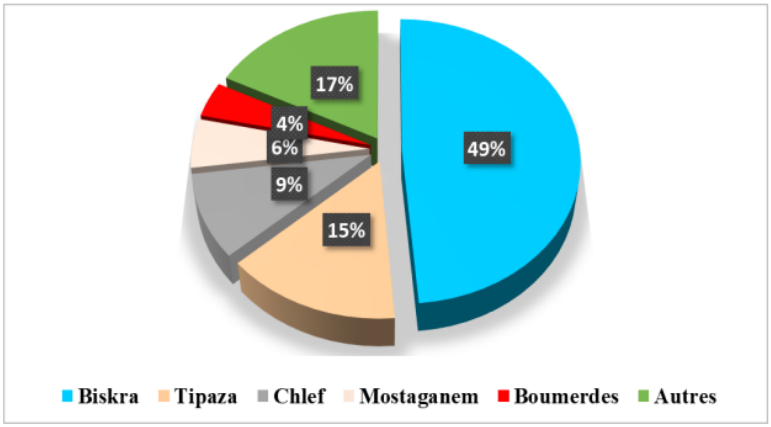


Figure 2.6: Tomato area in greenhouses by wilaya in 2016[22]

2.6.2.2 Tomato production in greenhouses by Wilaya

Biskra province led the production of fresh tomatoes in 2016, accounting for 26.39% of the national production, thanks to its favorable climate for tomato cultivation. Tipaza follows with 4.12%, utilizing heating systems in greenhouses. Chlef covers 3.34% of the national production of fresh products, while Algiers accounts for 1.56% of the national production.

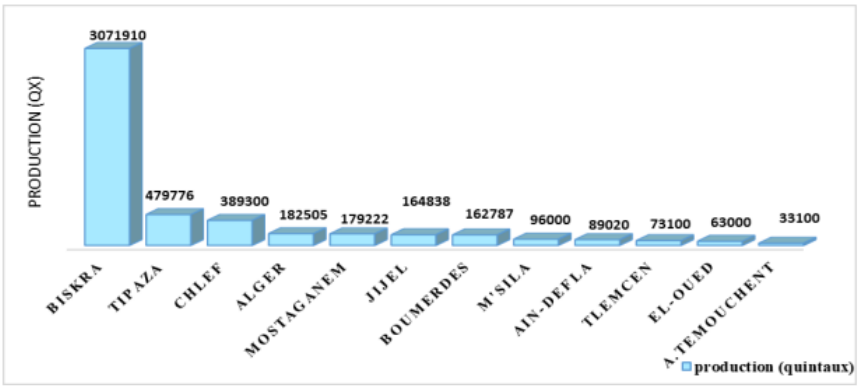


Figure 2.7: Tomato production in greenhouses by wilaya in 2016[22]

2.6.2.3 Greenhouse tomato yield by wilaya

In 2016, Biskra achieved a yield of 1403.3 quintals per hectare due to the application of intensive greenhouse agriculture and favorable pedoclimatic conditions for tomato cultivation, followed by Ain-Defla with a yield of 1290.1 quintals per hectare, and Jijel with a yield of 1216.9.

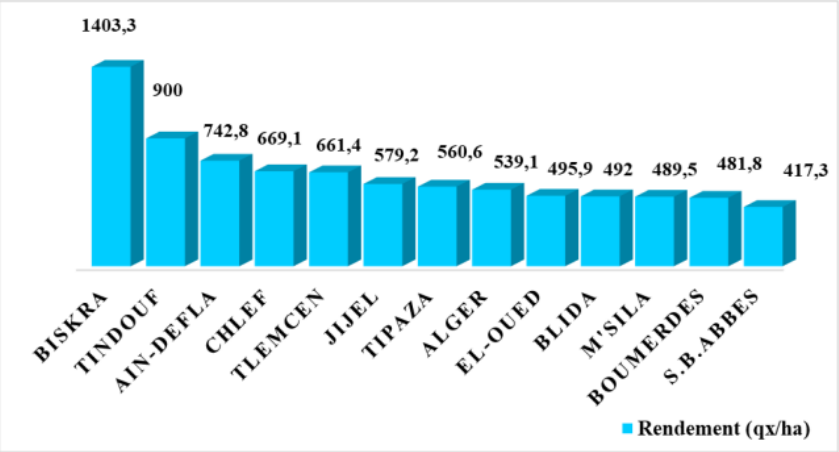


Figure 2.8: Greenhouse tomato yield by wilaya in 2016[22]

2.6.3 In Biskra

The Sahara and its oases are characterized by sandy soil and low precipitation (less than 25 mm in some regions). The state's support programs for the development of Saharan agriculture and the national agricultural development plan (PNDA) have eliminated climatic obstacles for farmers in the region. They are now encouraged to intensify vegetable cultivation in the southern provinces by the installation of wells and irrigation systems free of cost.As of 2002, vegetable cultivation in the area accounted for 11.02% of the national area, with production estimated at 12.24% of the national total. Today, it is the primary supplier of vegetables, particularly "extra early" varieties, to different regions in the country. Biskra province takes the lead among the 12 productive provinces in tomato production, with over 3 million quintals. El-Oued is the second-largest tomato-producing region with a production of over 1 million quintals, followed by Mostaganem in third place with a production of 939,128 quintals. Tipaza ranks fourth with a production of 848,514 quintals [91]. Biskra had the highest yield in 2016 at 1403 quintals per hectare, followed by Tindouf, Ain-Defla, and Chlef. After a decline in protected crops during the 1990s, much of the greenhouse infrastructure potential was transferred to the south, particularly to Biskra. Recently, there has been a surge of interest among greenhouse farmers, and subterranean crop cultivation has experienced significant growth, particularly with the introduction of multi-chamber greenhouses. Currently, the total area of greenhouses exceeds 7,000 hectares (2006), with 2,000 hectares located in the Biskra province, which had the highest yield in 2016.

2.7 Tomato Diseases: Overview and Classification

While there are many diseases and pests that can afflict tomato plants, so can we Divide it as follows:

2.7.1 Bacterial

Bacterial tomato disease is one of the most dangerous and devastating diseases affecting outdoor and greenhouse crops. The bacteria are carried by aerosols and rain spray and enter the leaves through stomata or wounds, where they multiply in the leaf. Symptoms of the disease include small black or brown necrotic lesions (spots) that may be surrounded by chlorotic halos. Lesions also form on unripe and ripe tomatoes, and this manifestation of the disease can reduce the product's marketability[98]. Bacterial spot in tomato causes severe yield and quality losses due to defoliation or lesions on the products[99].the disease on leaves shows small, irregular, dark lesions that coalesce and can cause general yellowing of the leaves. Spots and spots may appear on stems and petioles where they are indistinguishable. Infecting a flower with a bacterial spot can be a fairly serious infection, causing the flower to drop early[100].

Figure 2.9: Bacterial Spot in Tomato[23]

2.7.2 Insectes

Tomatoes are susceptible to many types of insects, each specializing in attacking a specific part of the plant (stem, products, leaf, etc.). The most common and widespread insect pests are whiteflies, which cause heat retention, Mizu persicae , and the Tuta absoluta. The products can become contaminated with whole insects, insect excrement, and insect parts [101]. Insects cause uneven ripening and a change in the internal color of products, and they may cause serious damage to the leaves, which reduces the plant's ability to undergo photosynthesis. This affects its growth and development[102].greenhouse tomato crops are greatly affected by the whitefly, which feeds on sap and secretes honeydew. These insects can cause sooty mold, a fungal disease that manifests as a blackish layer on the leaves. This condition can significantly impede photosynthesis and eventually suffocate the plants[103].

Figure 2.10: Whitefly on tomato leaf[18]

2.7.3 Virus

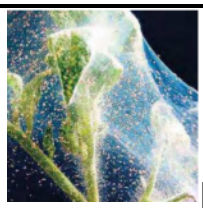
Viral diseases cause the biggest severe yield losses and that makes tomato production shift to greenhouses. These viruses include the tomato yellow leaf curl virus, tomato torrado virus, tomato spotted wilt virus, tomato chlorosis, tomato chlorosis, Pepino mosaic virus, and some small viruses[104]. The virus is spread by aphids or leafhoppers which can show leaf curl, a yellow or green mosaic pattern on leaves, leaving leaves "light" or tan. The products are also affected by mosaic patterns, streaks, or speckled areas[105]. The tobacco mosaic virus (TMV) is widespread worldwide and can cause significant damage in fields and greenhouses. TMV is one of the most stable viruses known. TMV can be transmitted through tomato seeds, is readily transmitted mechanically by human activity, and may be present in tobacco products. The virus is not spread by insects commonly found in greenhouses or fields[106]. Symptoms can be found at any stage of growth and all parts of the plant are affected. Usually, infected plants exhibit dark or light green spotting or mosaic with deformed young leaves and varying degrees of stunting. Severely affected leaves may have a fern appearance and may have raised dark green areas. Product sets can be severely reduced in affected trees [107].

Figure 2.11: Tobacco Mosaic Virus[24]

2.8 The most common diseases

2.8.1 Worldwide

Table 2.2: Worldwide common diseases

Disease	Symptomes	image
Early blight	characterized by dark and concentric lesions on the leaf or produit surface. Severe attacks often result in defoliation of the plant. The disease affects the quality, quantity and productivity of tomato produce[108].	 [108]
Septoria leaf spot	It first appears as small, water-soaked spots that soon become circular spots about an inch in diameter. The lesions gradually develop grayish-white centers with dark edges.fungal fruiting bodies appear as tiny black specks in the centers of the spots[109].	 [109]
Tomato Spider mites	Warm, dry conditions favor rapid spider mite population build-up and increased feeding which, causes severe losses tomato crops. the loss of leaf chlorophyll and the subsequent reduction in net photosynthetic rate causes leaf discoloration often called bronzing[110].	 [111]
Tomato White Mold	the white mold symptoms start as water-soaked lesions that rapidly expand and become necrotic lesions with intense white cottony fungal mycelium grown over them. White mold control is challenging because of its great capacity to survive. which can persist viable in the soil and crop debris for many years[112].	 [112]
Tomato Mosaic Virus (Tomv)	infected plants have a light or dark green mottling or mosaic with distortion of younger leaves, and stunting to varying degrees. Severely affected leaves may have a “fernlike” appearance and may show raised dark green areas[113].	 [113]

2.8.2 In Algeria

Table 2.3: The most common diseases in Algeria

Disease	Symptomes	image
Tuta Absoluta	The invasion of an exotic and rapidly spreading pest known as tomato leaf miner threatens tomato production. it causes heavy losses in tomato production. it can attack different parts of the host plant. the larva is the most dangerous stage that usually affects plant leaves but can also be found in produits and stems where [114].	 [114]
Botrytis Cinerea	The fungus often invades senescent or damaged plant tissue. Since the fungus is able to cause disease in plants at temperatures down to 2°C, it causes many problems in postharvest chains. To establish successful infection on healthy, undamaged tissue [115].	 [115]
Tomato Yellow Leaf Curl Virus (TYLCV)	Plants are severely stunted with shoots becoming erect. Leaflets are reduced in size and pucker. Leaflets curl upwards, become distorted, and have prominent yellowing along margins and/or interveinal regions. Flowers wither. Plants will set very few produit after infection occurs; therefore any plants infected before the flowering stage will produce extremely low yields [113].	 [113]

2.9 Detection Methods for Tomato Diseases

2.9.1 Traditionnel

The techniques to diagnosis of plant diseases are improving for long time. there is many facilities and technologies are developed and introduced to ensure diagnostic reliability and speed. traditional diagnostic methods make it possible to diagnose plant diseases in a timely the most importent methodses are:
Visual observation Visual observation is started from the place where the plant is grown, in which the diseased plant is compared with a healthy plant in plant height, color, leaf shape, leaf density on the branches, changes in the root system, and more. If there are foci of disease (necrosis or spots on the stems, leaves, etc.), the uses of magnifying glass to cheeck presence of mycelium, sporangiophora or sclerotia Microscopy microscopes are an important and widely used instrument For an accurate diagnosis high-precision.
for microscopic the infected parts of the plant are brought to the laboratory Mycological diagnosis In

the 'moist chamber' method, sick plant parts are placed in a high-humidity chamber (Petri dishes, etc.) and incubated. In this case, due to favorable conditions, fungi in the infected tissues develop and begin to manifest themselves. Biological assays or indicator plant tests. This method is widely used in the detection of phytoviruses and phytoplasmas, along with other diagnostic methods. Biological diagnosis is performed on indicator trees and herbaceous plants. Mechanical inoculation or injection is used to transmit the infection to indicator plants. In this case, the leaves with symptoms of the disease are rubbed into a healthy indicator plant, or a standard buffer[116].

2.10 Imaging Techniques and Visual Inspection

Diseases and pests are one of the factors that determine the quality of the product. image processing is a way to identify plant diseases and pests[117]. DL methods have recently broken into many areas of scientific research Among them agricultural and farming applications after being successfully used in various fields. which can help farmers to manage their crops more efficiently by Automating the identification of plant diseases, This is to increase the agricultural product. Detecting plant diseases on crops using images is a difficult task. In addition to their detection, the identification of individual species is necessary to apply appropriate control methods. A survey of research initiatives using convolutional neural networks which is a type of DL[118].

2.11 Challenges in Tomato Disease Detection

The future development of DL technology still poses various challenges and obstacles encouraging its greater use towards smarter, more sustainable and safer agricultural productivity[119]. Accurate identification and classification of plant diseases is essential to the well-being of agriculture. Manual disease detection in crops involves various barriers as significant efforts in terms of cost, manpower and expertise are required to accurately identify crop diseases. In addition, considering factors such as farming scale, different types of diseases and similar symptoms of different diseases, it is difficult for farmers to accurately and timely detect these diseases, which also affects negatively affected the yield. such as the quality of the crop[120]. The CNN samples are limited in their ability to process the original image data in its unstructured form.Deep learning systems require system engineering and design expertise to extract features from unstructured data into a feature vector through which subsystems can typically detect and classify feature's specific patterns in the input data. One of the ways to improve the accuracy of CNN models is to use large datasets so that the machine can improve its learning rate[121] The training images used must contain different samples taking into account different environments as well as lighting parameters. However, increasing the dataset size also increases the storage cost as well as the computational penalty[120].

2.12 Related Works

In[122]created a CNN model that gives 99.18% accuracy trained with 14828 images of tomato leaves infected with nine diseases. In [123] 2018, Belal A. M. Ashqar, and Samy S. Abu-Naser trained a deep convolutional neural network with a dataset of 9000 images containing infected and healthy Tomato leaves in specially controlled conditions the accuracy given 99.84% whene they trained model demonstrating the feasibility of this approach. In [124], Mohit Agarwal et al. talks about disease detection and classification using a CNN-based approach. the model that we created is a collection of three convolution layers and three pooling

layers. The classification accuracy varies from 76% to 100%for for the 9 diseases and 1 healthy class in 2020 India. A deep convolutional neural network was trained using the PlantVillage dataset with 14,903 images of diseased and healthy leaves The trained model achieves test accuracy of 99.25% the work is done by Akshay Kumar al [125]. As early as 2020 Murk Chohan al [126] Use a dataset of 54,306 images of diseased and healthy leaves collected under controlled conditions, the DCNN is trained to identify not just tomato plants but 14 crop species and 26 diseases The trained model shows an excellent result with an accuracy of 99.35% on a held-out test set.YANG ZHANG al [127] decide to develop tomato disease detection methods based on deep convolutional neural networks and object detection models. Two different models, Faster R-CNN and Mask R-CNN, are used in these methods, where Faster R-CNN is used to identify the types of tomato diseases, and Mask R-CNN is used to detect and segment the locations and shapes of the infected areas.

2.13 Conclusion

In conclusion, plant diseases pose a significant threat to agriculture and the economy. They can cause a decline in production quantity and quality, leading to significant economic losses. It is crucial to identify and manage plant diseases promptly to minimize their impact on agriculture and the environment. Timely diagnosis and appropriate management practices, including crop rotation, use of resistant plant varieties, and appropriate pesticide application, are essential in preventing the spread of plant diseases. Furthermore, the impact of plant diseases on human health cannot be overlooked, and proper food safety measures must be taken to prevent the consumption of contaminated produce.

Chapter 3

Conception and Implementation

3.1 Introduction

In our project, our goal is to build a convolutional neural network that can accurately detect and classify tomato diseases. To achieve this, we need to design a model that delivers the best possible results. This chapter presents the general and detailed design of the proposed system. Following that, we will cover implementation details, including the necessary software and hardware requirements to build the system. We will also describe the process of building the model and discuss the steps required to prepare the data for training.

3.2 Global Architecture of the system

To develop a program that accurately identifies tomato plant diseases, some steps must be followed. The first step was a large collection of data including diseases that show symptoms on leaves and products, and the second step involves implementing different pre-processing techniques on the images to obtain a uniform size fit for the CNN model. This involves applying transformations such as rotate, crop, flip, and scale to create new images with subtle differences that can help the model's ability to generalize, to train the software to accurately identify these diseases. Figure 1 shows some of these steps.

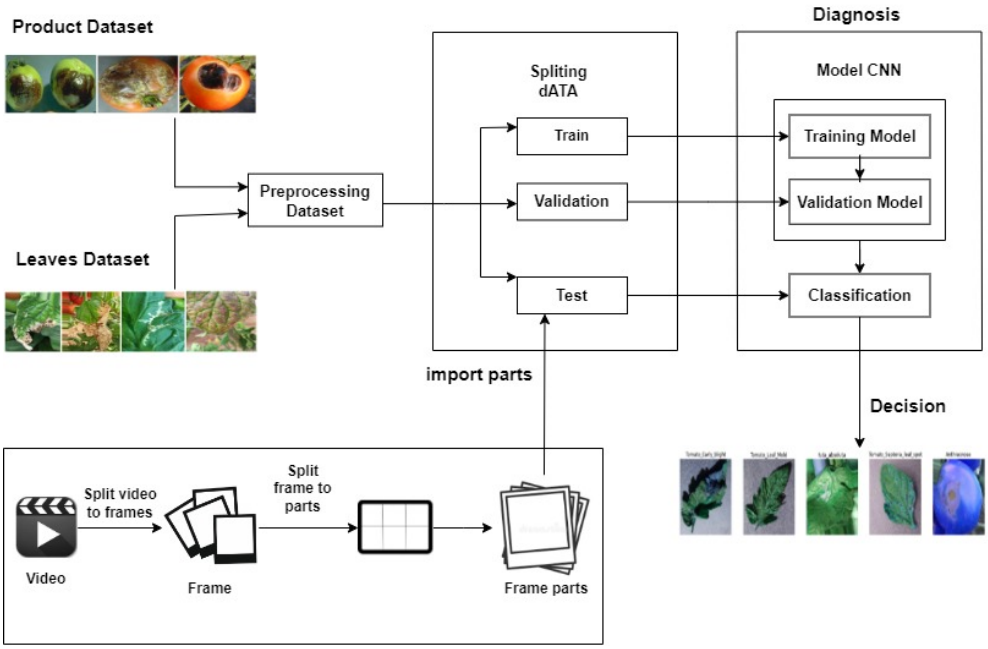


Figure 3.1: The General Architecture of Our System.

3.3 Detailed Architecture

In this section, we will provide a comprehensive overview of our work, which includes data collection, as well as a detailed description of each step accompanied by diagrams for better visualization 3.2.

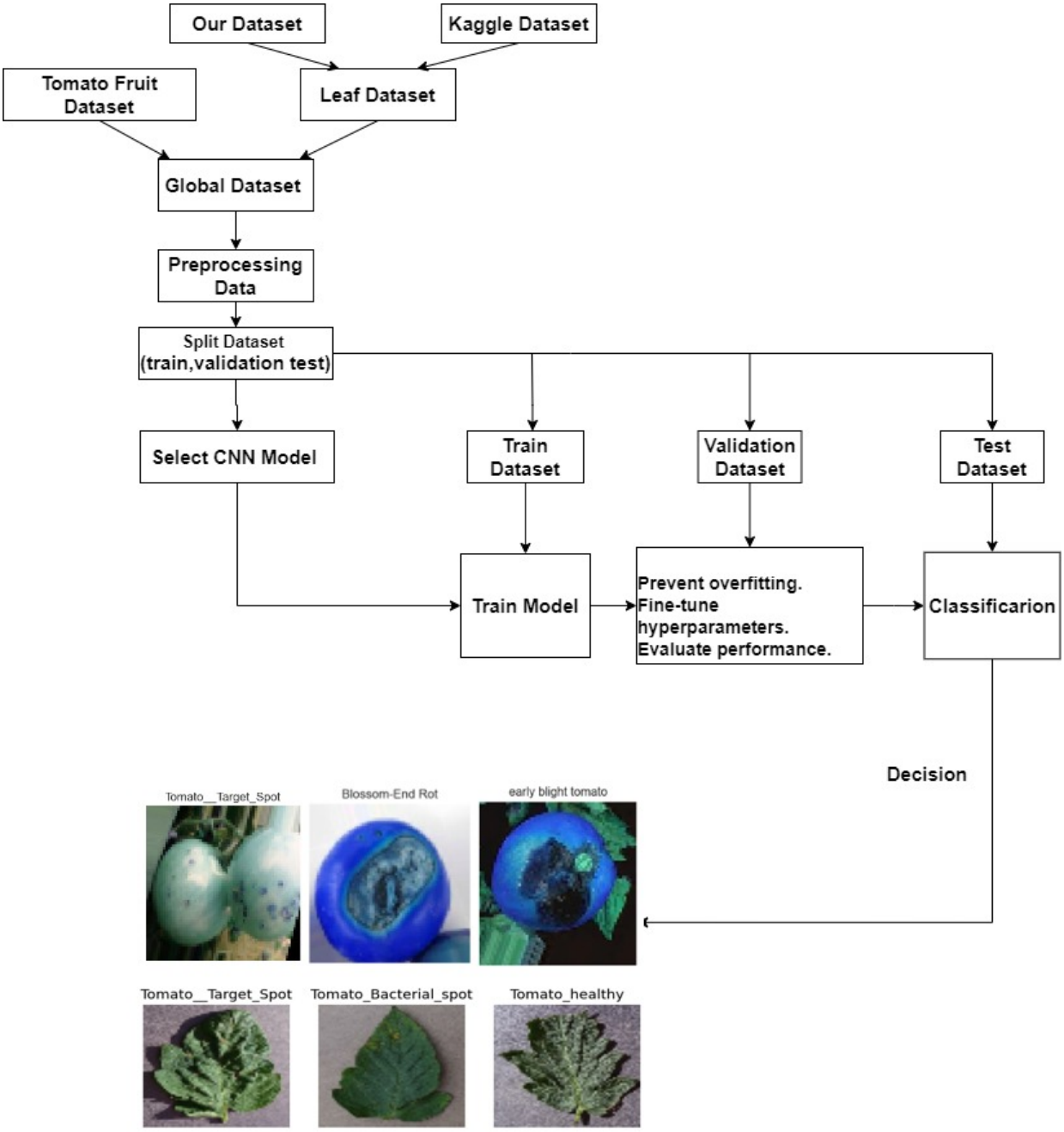


Figure 3.2: Detailed Architecture of Our System.

3.4 Dataset

In order to train the model effectively, it was essential to gather an appropriate dataset first. We started by exploring the available datasets related to our problem domain, and then we decided to create our own dataset that reflects the most common environmental conditions around us, with the help of the available datasets.

3.4.1 Collect Dataset

In this study, two types of datasets were utilized - one that focused on leaves and the other on tomato produits.

3.4.1.1 Leaf dataset

The data of the leaves used is divided into two parts: The first section is the data downloaded from the site and is available on kaggle[24]. The training set contains nearly 15,000 images of tomato leaves categorized into 10 categories. Each category expresses a disease that affects tomatoes in its own leaves. As for the second dataset, it was collected and photographed by us, where we went to several centers to search for a group of images and consult specialists in plant diseases in order to determine the most common diseases in the state of Biskra and to take comprehensive knowledge on the subject and among the institutions that were directed to it is the Agricultural Center desert and the Plant Protection Center, where some information was collected, then they were taken to one of the plastic houses located in the municipality of Muzaira'a, and about 3000 pictures of various diseases were photographed, and then the diseases were classified according to the symptoms present in each leaf, which was a difficult work as it Some diseases are different in the advanced stages over the later stages in relation to the apparent symptoms, and for us, and because we lack experience, the subject was very difficult, and because the accuracy of the model depends on the correct classification, after the images were classified, the help of the National Plant Protection Center was sought to determine each disease and its name, With symptoms of nutritional deficiency, not necessarily all diseases.

3.4.1.2 Product dataset

Our goal was to collect images of tomato produits affected by different diseases to create a dataset, which led us to relocate to the Muzaira'a municipality, after searching for an available and ready data set collection that was not available after going to the tomato trees in one of the greenhouses, we found that the tomatoes were not fully ripe and that all diseases The spread affected only the leaves, but the tomatoes did not, and there we were told that it is difficult to find diseased tomato produits in any plastic house because most farmers preserve the crop for fear of spoilage, so whoever wants to collect pictures of data, he can plant a tomato tree and then expose it to each disease so that it appears Symptoms, then pictures are taken for the symptoms, and because this solution may take time and requires a lot of experience in cultivation and caring for the plant, and there is another solution, which is collecting pictures from various sources, books, articles, websites, all of which are concerned with all tomato diseases, As a result, we had to rely on on three data collection sites, which contained images of ten different types of tomato diseases, to create a comprehensive dataset [128][129].

3.4.2 Preprocessing Dataset

Preprocessing is a crucial step in machine learning that involves converting a raw dataset into a usable format for models. It is necessary to reduce dimensions, extract relevant data, and enhance the performance of certain machine learning models. The process entails transforming and encoding data to make it easily interpretable by computers.

3.4.3 Data Labeling

We classified the images based on the apparent symptoms and subsequently sought the expertise of a group of specialists in plant diseases to identify and classify the existing diseases accurately.

3.4.4 Data Spliting

The dataset was split into training, testing and validation sets, we train the model using 80% of data and 20% for testing. Splitting was done manually to ensure that the segmentation of the data represented the distribution of diseases in the entire dataset.

3.4.5 Data Resize

Resizing images is a critical step in preprocessing for computer vision, allowing deep learning models to train faster. However, using larger images as input results in longer training times as the neural network has to learn from four times as many pixels. Resizing images to a smaller size can therefore significantly reduce the training time and improve the performance of the model. It is also important to ensure uniform image size throughout the dataset. Our dataset ranges in size from 224 to 512, along with other produit datasets and Dataset Kaggle. To standardize and scale the dataset, we resized the images to 150.

3.4.6 Data Augmentation

Data augmentation is a technique used to generate additional training data by applying various transformations to existing data to obtain new data. However, since the data extracted from Kaggle was already large and prepared, we did not apply data augmentation to it. Instead, we applied data augmentation to the rest of the dataset.

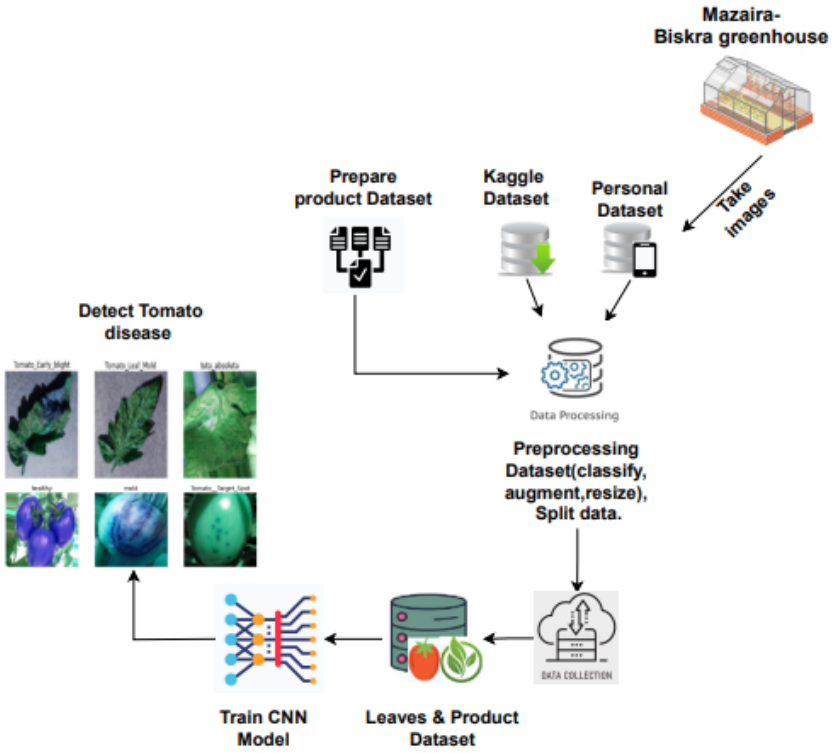


Figure 3.3: Dataset processing stages.

3.5 Application architecture

To streamline the modal usage model, it is recommended to provide dedicated mobile applications. which has summarized the most important operations that can be applied in the following architecture.

3.6 Implementation of the model in IoT system.

The following structure shows how to develop a system responsible for detecting tomato plant diseases inside greenhouses.

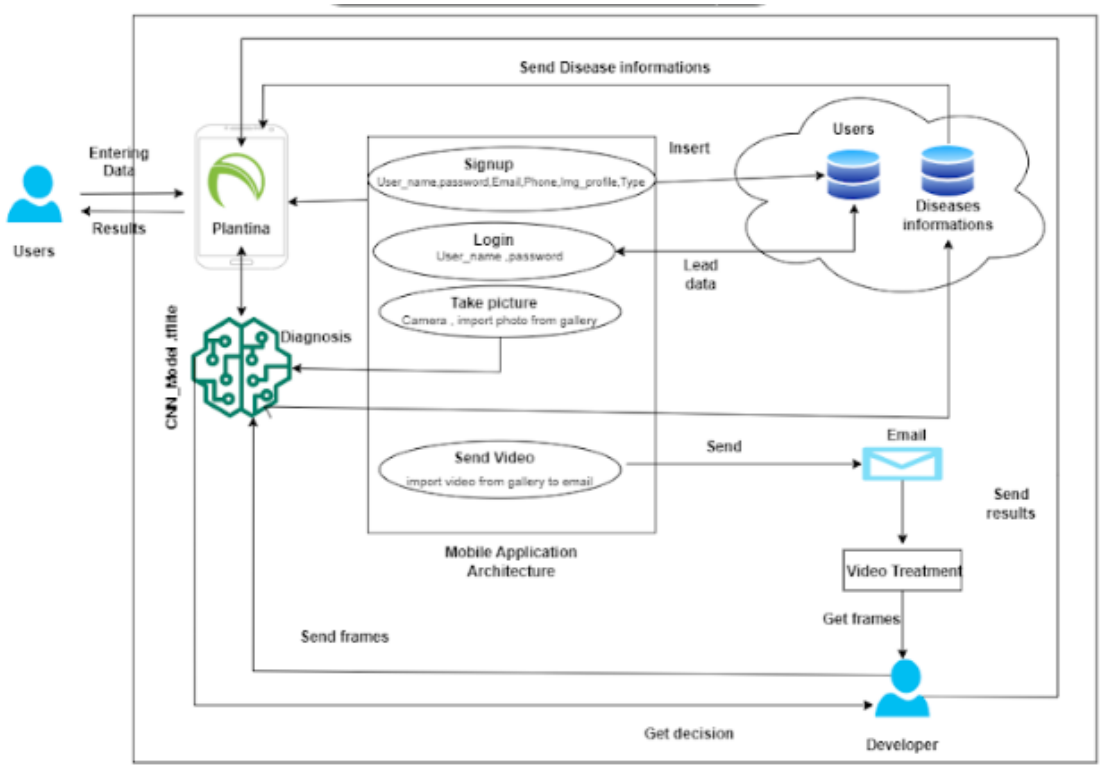


Figure 3.4: Application architecture.

3.7 CNN learning

Table 3.1: CNN layers parameters

	Description
Layers	CNNs are comprised of three types of layers. These are convolutional layers, pooling layers, and fully-connected layers. When these layers are stacked, a CNN architecture has been formed[130].
Optimizers	Optimizers in deep learning are algorithms that adjust the weights and biases of neural networks during training in order to minimize the error function and accelerate convergence to an optimal global solution. These algorithms work by updating the parameters in small increments based on the gradient of the loss function with respect to each parameter, and they are typically used in stochastic gradient descent-based optimization algorithms[131].
Loss Function	The loss function is the function that computes the distance between the current output of the algorithm and the expected output. It's a method to evaluate how your algorithm models the data. It can be categorized into two groups. One for classification and the other for regression[132]

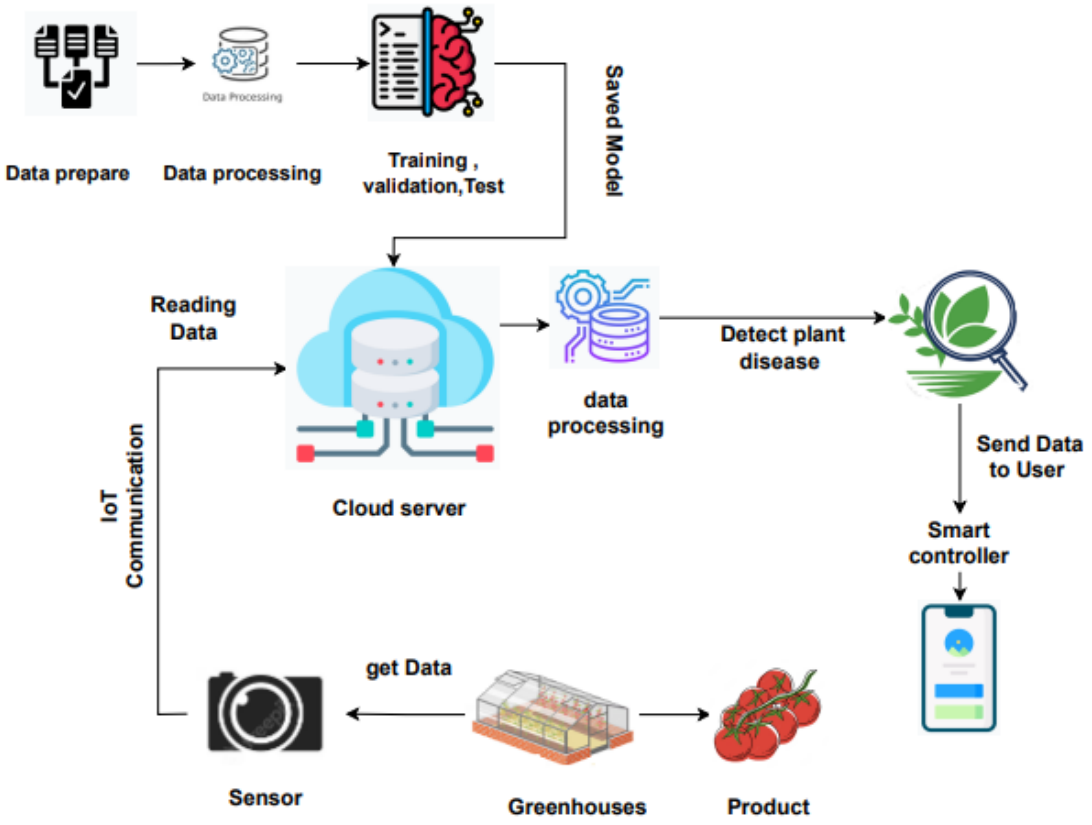


Figure 3.5: IoT system for tomato diseases monitoring.

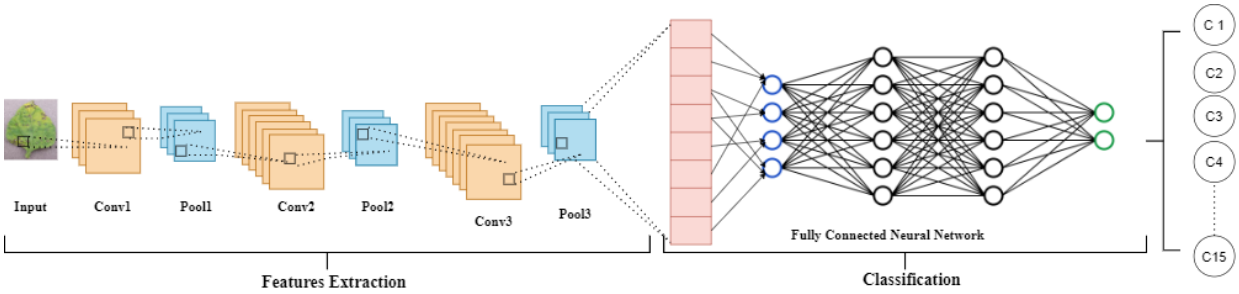


Figure 3.6: Our proposed CNN architecture.

3.7.1 Training phase of our Model

After the organization all data and splitting them into three sections training data, validation data, and testing data which contain JPG images the next step is to train our CNN architecture and fit the training images being labeled with their corresponding class labels in many epochs to find the best weights and save them. After training the CNN, we use validation data to verify its accuracy. The validation data is a set of labeled images not included in the training process. Essentially, the test dataset evaluates the prediction performance of the CNN.

3.7.2 Classification phase

Once the training phase is complete, the CNN model is prepared for image classification during the prediction phase. This involves feeding new input data to the model, which consists of the third set of sections which is test set. The model is trained in this phase to assess its accuracy and ability to correctly classify images.

3.8 CNN model Evaluation

3.8.1 Confusion Matrix

The confusion matrix is a tabular representation that displays the number of occurrences between the true/actual classification and the predicted classification by the model, as shown in Figure 1. The matrix has two dimensions, where the columns represent the predicted classes, and the rows represent the true classes. To maintain consistency throughout the analysis, the order of the classes in the rows is the same as in the columns. The correctly classified elements are located on the main diagonal, from the top left to the bottom right, indicating the number of times the predicted class agrees with the true class[133].

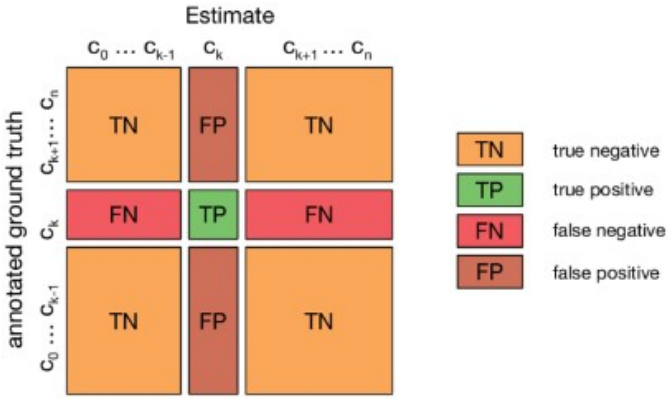


Figure 3.7: Confusion matrix for multi-class classification [25].

The confusion matrix of a classification with n classes. When considering the class k ($0 \leq k \leq n$)

3.8.2 Evaluation of the CNN model

In this study, we examined the performance of the classification model for tomato leaf disease identification based on the CNN model. The performance of each classifier was measured in terms of accuracy, recall, precision, and F1 Score. The confusion matrix measures are expressed in Eqs[134].

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

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TP}{TP + FP}$$

$$FPR = \frac{FP}{FP + TN}$$

$$F1Score = 2 * \frac{sensitivity * precision}{sensitivity + precision}$$

where, TP = true positive, TN = true negative, FP = false positive, FN = false negative. Calculating the accuracy is not enough for performance evaluation especially when the dataset is unbalanced it is essential to measure the f1 score we can say that model is strong if the f1 score is close to 1.0 which is the weighted average of the correct score and the recall score[135]. Precision (PRE) and recall (REC) are more commonly used metrics in information technology and relate to false positive and true positive rates. A recall is synonymous with a true positive rate and is also sometimes called sensitivity. In another way, the precision measures the ability of the model to avoid the false positives [136].

3.9 Implementation of our deep learning model

3.9.1 Hardware configuration

The work has been done using two laptops which are characterized by :

3.9.1.1 LENOVO ideapad 320

— Processor : Intel(R) Core(TM) i3-6006U.
— Processor Frequency: 2.00 GHz.
— RAM: 12 Go.
— Hard drive: 1TB HDD.

3.9.1.2 ACER Aspire E5-573

— Processor : Intel(R) Core(TM) i3-5005U.
— Processor Frequency: 2.00 GHz.
— RAM: 12 Go.
— Hard drive: 500 Go HDD.

3.9.2 Development environment

The development of this system needs a collection of tools and environments as shown in the table below :

Table 3.2: Development environment

Logo	Description
	Python is a powerful, interpreted, interactive, object-oriented, high-level, object-oriented scripting language created by Guido Van Rossum in the late 1980s. Python is a language well-suited for beginner programmers. and support the development of many applications ranging from simple. handle for www browsers and game development. One of the main reasons for Python's rapid growth is the simplicity of its syntax. The language reads almost like regular English, making it easy to write complex programs[137].
	In 2012, Anaconda was developed by Peter Wang and Travis Oliphant in response to the growing demand for Python in business data analytics. With the emergence of new technology trends, the field was rapidly transforming. At the same time, there was a need for an entity to organize and unify the open-source community to harness its full potential[138].
	The Jupyter Notebook is a tool available in open-source and can be accessed via a web browser, serving as a digital laboratory notebook that aids in managing workflows, code, data, and visualizations, and documents the research process. This tool is both machine and human-readable, enabling seamless interoperability and fostering scholarly communication[139].
	Android Studio is the official integrated development environment (IDE) for Android application development. It is based on IntelliJ IDEA, a Java-integrated development environment for software, and incorporates its code editing and developer tools[140].
	Firebase is the name of a mobile platform from Google that facilitates the creation of back-ends that are both scalable and efficient. In other words, it is a platform that allows the rapid development of applications for mobile and for the web offering good flexibility of widespread use[141].

3.9.3 Packages and APIs

Many packages and APIs have been used to create the CNN model, which is very useful to define any functionality. The table below shows them:

Table 3.3: Packages and APIs

Logo	Description
TensorFlow	TensorFlow is an open-source framework that has been created by Google researchers to run statistical and predictive analytics workloads, including machine learning and deep learning. It has been designed to simplify the process of developing and executing advanced analytics applications for various users such as statisticians, data scientists, and predictive modelers[142].
Keras	Keras is a Python-based deep learning API that runs on top of the TensorFlow machine learning platform. It was designed to facilitate rapid experimentation by emphasizing speed and ease of use. The ability to quickly move from concept to outcome is essential for conducting effective research[143].
NumPy	NumPy is a Python library for numerical computing that provides support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. It was first released in 2006 and was built upon earlier efforts of the Numeric and Numarray libraries [144].
OpenCV	OpenCV is a free and open-source computer vision and machine learning software library. It was developed with the aim of providing a unified infrastructure for computer vision applications and accelerating the adoption of machine perception in commercial products. OpenCV is distributed under the Apache 2 license, which makes it easy for businesses to use and modify the code according to their needs[145].
matplotlib	Matplotlib is a data visualization library that can be used across multiple platforms, built on NumPy arrays and designed to work seamlessly with the broader SciPy stack. It was initially introduced by John Hunter in 2002. One of the main advantages of data visualization is that it enables us to get visual access to large amounts of data through easily digestible visuals. Matplotlib includes various types of plots, such as line, bar, scatter, histogram, and many more[146].

3.9.4 Dataset Preparation and Preprocessing

3.9.4.1 Data augmentation

We utilized this method on both our dataset and the produits dataset to increase the dataset's size, thus enhancing the model's ability to generalize to new, unseen data Figure3.8.

```python
from keras.preprocessing.image import ImageDataGenerator
from skimage import io
datagen = ImageDataGenerator(rotation_range=20,
                             width_shift_range=0.1,
                             height_shift_range=0.1,
                             shear_range=0.2,
                             zoom_range=0.2,
                             horizontal_flip=True,
                             fill_mode='nearest')
import numpy as np
import os
from PIL import Image
image_directory = r'/path/to/input/directory'
SIZE = 224
dataset = []
my_images = os.listdir(image_directory)
for i, image_name in enumerate(my_images):
    if (image_name.split('.')[1] == 'jpg'):
        image = io.imread(image_directory + image_name)
        image = Image.fromarray(image, 'RGB')
        image = image.resize((SIZE,SIZE))
        dataset.append(np.array(image))
x = np.array(dataset)
i = 0
for batch in datagen.flow(x, batch_size=16,
                          save_to_dir= r'/path/to/output/directory',
                          save_prefix='dr',
                          save_format='jpg'):
    i += 1
    if i > 50:
        break
```

Figure 3.8: Illustration of Dataset augmentation.

3.9.4.2 Data resize

In this step, we standardized the size of all images to expedite the model training process. This is because the fact that the dataset images are large does not provide a boost to CNN performance, and often results in longer training times, as show in Figure 3.9.

```python
from PIL import Image
import os

# set the directory path where the images are located
directory = 'G:/code/Dataset/testing_set/move/'

# set the output directory where the resized images will be saved
output_directory = 'G:/code/Dataset/testing_set/move/'

# set the desired output size for the images
output_size = (224,224)

# loop through all the images in the directory and resize them
for filename in os.listdir(directory):
    if filename.endswith('.jpg') or filename.endswith('.png'): # check for image file extensions
        # open the image file
        img = Image.open(os.path.join(directory, filename))

        # resize the image
        resized_img = img.resize(output_size)

        # save the resized image to the output directory with the same filename
        resized_img.save(os.path.join(output_directory, filename))
```

Figure 3.9: Illustration of Dataset Resize.

3.9.4.3 Split Dataset

This code is use to split a dataset into train, validation, and test sets using the train_test_split() function and creates subfolders for each set.

```
import os
import shutil
from sklearn.model_selection import train_test_split

data_dir = 'G:/dataof artivle/train1'
output_dir = 'G:/dataof artivle/'

train_dir = os.path.join(output_dir, 'train')
val_dir = os.path.join(output_dir, 'val')
test_dir = os.path.join(output_dir, 'test')

os.makedirs(train_dir, exist_ok=True)
os.makedirs(val_dir, exist_ok=True)
os.makedirs(test_dir, exist_ok=True)

for folder in os.listdir(data_dir):
    folder_path = os.path.join(data_dir, folder)
    files = os.listdir(folder_path)
    train_files, val_test_files = train_test_split(files, test_size=0.3, random_state=42)
    val_files, test_files = train_test_split(val_test_files, test_size=0.33, random_state=42)

    # Create subdirectories for each folder in train, validation, and test directories
    os.makedirs(os.path.join(train_dir, folder), exist_ok=True)
    os.makedirs(os.path.join(val_dir, folder), exist_ok=True)
    os.makedirs(os.path.join(test_dir, folder), exist_ok=True)

    # Move files to the corresponding directories
    for file in train_files:
        src_path = os.path.join(folder_path, file)
        dst_path = os.path.join(train_dir, folder, file)
        shutil.copy(src_path, dst_path)
    for file in val_files:
        src_path = os.path.join(folder_path, file)
        dst_path = os.path.join(val_dir, folder, file)
        shutil.copy(src_path, dst_path)
    for file in test_files:
        src_path = os.path.join(folder_path, file)
        dst_path = os.path.join(test_dir, folder, file)
        shutil.copy(src_path, dst_path)
```

Figure 3.10: Split Dataset code.

3.9.4.4 Preparing Training set

In this code, we are reading images from the training set and saving them as a list of NumPy arrays in 'X_train'. The corresponding categories of each image are stored in 'y_train' as a list.

```
X_train = []
y_train = []
i=0
for folder in  os.listdir(trainpath ) :
    files = gb.glob(pathname= str( trainpath + folder + '/*.jpg'))
    for file in files:
        image = cv2.imread(file)
        if image is not None:

         image_array = cv2.resize(image , (s,s))
         X_train.append(list(image_array))
         y_train.append(code[folder])

print('y',len(y_train))
print(f'we have {len(X_train)} items in X_train')
```

Figure 3.11: Preparing Training Dataset.

3.9.4.5 Preparing Validation set

like the previous step we read the validation set and save them as a list of NumPy arrays in 'X_test'.

```
X_test = []
y_test = []
for folder in  os.listdir(testpath ) :
    files = gb.glob(pathname= str(testpath +folder + '/*.jpg'))
    for file in files:
        image = cv2.imread(file)
        image_array = cv2.resize(image , (s,s))
        X_test.append(list(image_array))
        y_test.append(code[folder])
print(f'we have {len(X_test)} items in X_test')
```

Figure 3.12: Preparing Validation Dataset.

3.9.4.6 Preparing Testing set

For preparing testing set , this code reads all the images in a test set and resizes them to a uniform size before storing them as a list of numpy arrays in 'X_pred'.

```
X_pred = []
files = gb.glob(pathname= str(predpath + '/*.jpg'))
for file in files:
    image = cv2.imread(file)
    image_array = cv2.resize(image , (s,s))
    X_pred.append(list(image_array))
print(f'we have {len(X_pred)} items in X_pred')
```

Figure 3.13: Preparing Testing Dataset.

3.9.5 Building Our CNN Model

3.9.5.1 Import libraries and modules

To construct a CNN model, the initial step involves importing the required libraries. These libraries provide the necessary tools and functions for creating and training convolutional neural networks.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="whitegrid")
import os
import glob as gb
import cv2
import tensorflow as tf
import keras
from sklearn.metrics import confusion_matrix
from sklearn import tree
from sklearn import datasets
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import precision_score
```

Figure 3.14: Necessary libraries for building a CNN model.

3.9.5.2 Creating CNN model

The table 3.4 presented below provides a comprehensive description of the various layers within a deep convolutional neural network. It outlines the specific configurations and specifications for each layer present in the network.

Table 3.4: The different layers of DCNN-TD.

	Kernel			
	size	numbre	Input	Output
Conv 1	3*3	128	150*150*3	150, 150, 128
Pool 1	2*2		150*150*182	37, 37, 128
Conv 2	3*3	256	37*37*128	37, 37,256
Pool 2	2*2		37*37*256	9, 9, 256
Conv 3	3*3	512	9*9*256	9, 9, 512
Pool 3	2*2		4*4*512	4, 4, 512
FCL 1				128
FCL 2				64
FCL 3				32

The parameter shown in the table are describing our model that does a multi-classification of the diseases which gives the best accuracy and loss results using the Adam optimizer:

Table 3.5: CNN Training Parameters.

Parameter	product Dataset	leaves Dataset
Input shape	(150,150)	(150,150)
epochs	15	50
Batch size	64	64
Layer Activation	ReLU	ReLU
Dence Activation	SoftMax	SoftMax
Time(epoch/s)	77.6	596.6
Optimizer	adam	adam

```
KerasModel =keras.models.Sequential([
        keras.layers.Conv2D(128,kernel_size=(3,3),padding="same", activation='relu',input_shape=(s,s,3)),
        keras.layers.MaxPool2D(4,4),
        keras.layers.Conv2D(256,kernel_size=(3,3),padding="same",activation='relu'),
        keras.layers.MaxPool2D(4,4),
        keras.layers.Conv2D(512,kernel_size=(3,3),padding="same", activation='relu'),
        keras.layers.MaxPool2D(2,2),
        keras.layers.Flatten() ,
```

Figure 3.15: CNN model architecture.

The last layer is for the multi-classification which is a fully connected neural network :

```
keras.layers.Dense(units=128, activation='relu'),
keras.layers.Dense(units=64, activation='relu'),
keras.layers.Dense(units=32, activation='relu'),
keras.layers.Dense(units=26, activation='softmax'),
])
```

Figure 3.16: Illustration of classification model.

3.9.5.3 Compiling CNN Model

Once the model is built, the next step is to configure it using the compile() method of the Model class. This method is used to set the optimizer, loss function, and metrics to be used during training.

```
KerasModel.compile(optimizer ='adam',
                   loss='sparse_categorical_crossentropy',metrics=['accuracy'])
```

Figure 3.17: Compiling model.

3.9.5.4 Model summary

to see the model details we can use summary() method to display its contents the table shows all the parameters of the model: 3.19

```
KerasModel.summary()
```

Figure 3.18: Model Summary.

```
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 conv2d_12 (Conv2D)           (None, 64, 64, 128)       3584

 max_pooling2d_12 (MaxPoolin  (None, 16, 16, 128)      0
 g2D)

 conv2d_13 (Conv2D)           (None, 16, 16, 256)       295168

 max_pooling2d_13 (MaxPoolin  (None, 4, 4, 256)        0
 g2D)

 conv2d_14 (Conv2D)           (None, 4, 4, 512)         1180160

 max_pooling2d_14 (MaxPoolin  (None, 2, 2, 512)        0
 g2D)

 flatten_4 (Flatten)          (None, 2048)              0

 dense_16 (Dense)             (None, 128)               262272

 dense_17 (Dense)             (None, 64)                8256

 dense_18 (Dense)             (None, 32)                2080

 dense_19 (Dense)             (None, 26)                858

=================================================================
Total params: 1,752,378
Trainable params: 1,752,378
Non-trainable params: 0
```

Figure 3.19: Illustration of our model summary.

3.9.5.5 Training CNN model

When the model is built we can use fit to train our model:

```
ThisModel = KerasModel.fit(
    X_train, y_train,
    epochs=50,
    batch_size=64,
    validation_data=(X_test, y_test)
)
```

Figure 3.20: Training model.

firstly we must feed the model by training data (X-train) and their results (Y-train) then the validation by(X-test,Y-test) repeating the operation in 50 epochs on a batch size of 64

3.9.6 Testing our CNN model

After training a CNN model, we should test it to evaluate its performance on unseen data. To do this, we can use the following code:

```
y_result = KerasModel.predict(X_pred_array)

print('Prediction Shape is {}'.format(y_result.shape))

plt.figure(figsize=(10,10))
for n , i in enumerate(list(np.random.randint(0,len(X_pred),36))) :
    plt.subplot(6,6,n+1)
    plt.imshow(X_pred[i])
    plt.axis('off')
    plt.title(getcode(np.argmax(y_result[i])))
```

Figure 3.21: Testing model.

3.10 Extract Frames from Video

In order to streamline the disease identification process for the entire plant, we propose implementing a video clip-based identification feature. Here's how the process works:
1-Upon receiving a video, it is segmented into individual images at a frequency of one image per second.
2-Once we have the set of images, each image is further divided into six sections.
3-Each section is then passed through a model designed to identify any diseases present in that particular section.
4-After the identification is performed for all the sections, an analysis is conducted on the collective results to obtain a comprehensive assessment of the apparent diseases across the entire plant.
By employing this approach, we aim to enhance the efficiency and accuracy of disease identification, enabling prompt and effective management strategies for plant health.

```
from keras.preprocessing.image import ImageDataGenerator
from skimage import io
datagen = ImageDataGenerator(rotation_range=20,
                             width_shift_range=0.1,
                             height_shift_range=0.1,
                             shear_range=0.2,
                             zoom_range=0.2,
                             horizontal_flip=True,
                             fill_mode='nearest')
import numpy as np
import os
from PIL import Image
image_directory = r'C:/Users/MICRO/Desktop/healthy/'
SIZE = 256
dataset = []
my_images = os.listdir(image_directory)
for i, image_name in enumerate(my_images):
    if (image_name.split('.')[1] == 'jpg'):
        image = io.imread(image_directory + image_name)
        image = Image.fromarray(image, 'RGB')
        image = image.resize((SIZE,SIZE))
        dataset.append(np.array(image))
x = np.array(dataset)
i = 0
for batch in datagen.flow(x, batch_size=16,
                          save_to_dir= r'C:/Users/MICRO/Desktop/healthy',
                          save_prefix='dr',
                          save_format='jpg'):
    i += 1
    if i > 50:
        break
```

Figure 3.22: Split Video into Frames

```
import os
from PIL import Image
from itertools import product
def tile(filename, dir_in, dir_out, d):
    name, ext = os.path.splitext(filename)
    img = Image.open(os.path.join(dir_in, filename))
    w, h = img.size

    grid = product(range(0, h-h%d, d), range(0, w-w%d, d))
    for i, j in grid:
        box = (j, i, j+d, i+d)
        out = os.path.join(dir_out, f'{name}_{i}_{j}{ext}')
        img.crop(box).save(out)
tile('ss.png','D:/','D:/here',128)
```

Figure 3.23: Divide Image

3.11 Conclusion

In this chapter, we presented our dataset structures, described our system design, explained step by step the preprocessing phase, mentioned the tools, libraries, and frameworks we used, presented the implementation of a large part of our system and discussed our CNN model in detail. In the following chapter, we will go over various experiments and their outcomes.

Chapter 4

Results and Discussion

4.1 Introduction

In the previous chapter, we discussed our system architecture, demonstrated the structures of the used dataset, and provided a code source for each part of our classification system. In this chapter, We will explain our experimental results of the tomato plant disease classification in both cases leaves or product. Our results describe the obtained results for the multi-classification of tomato disease. also, we show the mobile application and all its functions.

4.2 Results and discussion

This study focuses on identifying diseases in tomato plants in Biskra, Algeria. It employs scientific technology, specifically deep learning, to detect diseases in greenhouses growing tomato plants at an early stage. The study uses the Python programming language on the Jupyter environment, along with Tensorflow and Keras libraries, to develop a programmed model that can recognize images using Convolutional Neural Networks (CNN). validation was conducted on 30% of existing images, which were two, leaves datasets and products dataset. The results of the study were encouraging, with an estimated accuracy rate of 98% (leaves dataset) and 99% (product dataset). The programmed model is capable of determining whether tomato leaves and products are healthy or not and can identify possible diseases.
Considering that our dataset is split into two distinct parts—one for leaves and the other for products, we trained separate models for each category. As a result, we obtained two sets of results that demonstrate the effectiveness of the developed models. The performance metrics for both leaf and product models are presented below, showcasing the merit of our approach.

4.2.1 leaves Dataset Results

Below are the results obtained from training the model on the dataset of leaves, which is categorized into 16 classes:

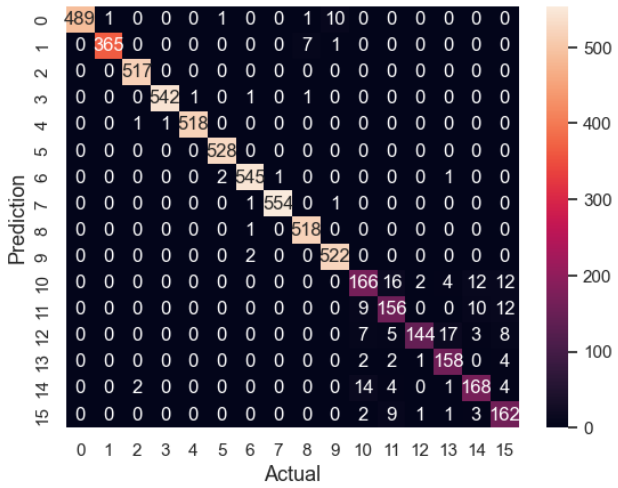


Figure 4.1: Confusion matrix of leaves Dataset.

Analysis of the Results The experimental findings shed light on the training progression of the model across 50 epochs. By examining the loss and accuracy values, several noteworthy trends and patterns emerge.

As shown in Figure 4.2 and 4.3 below, During the initial epoch, the model exhibited a loss of 2.0958 and an accuracy of 0.4723. However, as the training advanced through the 50 epochs, the model consistently demonstrated a reduction in loss and an improvement in accuracy, indicating its ability to make more accurate predictions. Ultimately, the training concluded with an impressive decrease in the loss value to 0.0634, coupled with a notable increase in accuracy to 0.9813.

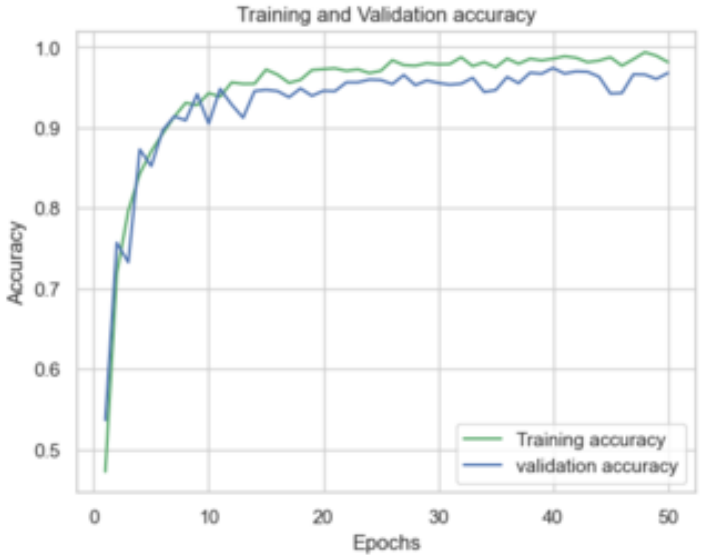


Figure 4.2: Average Accuracy curve.

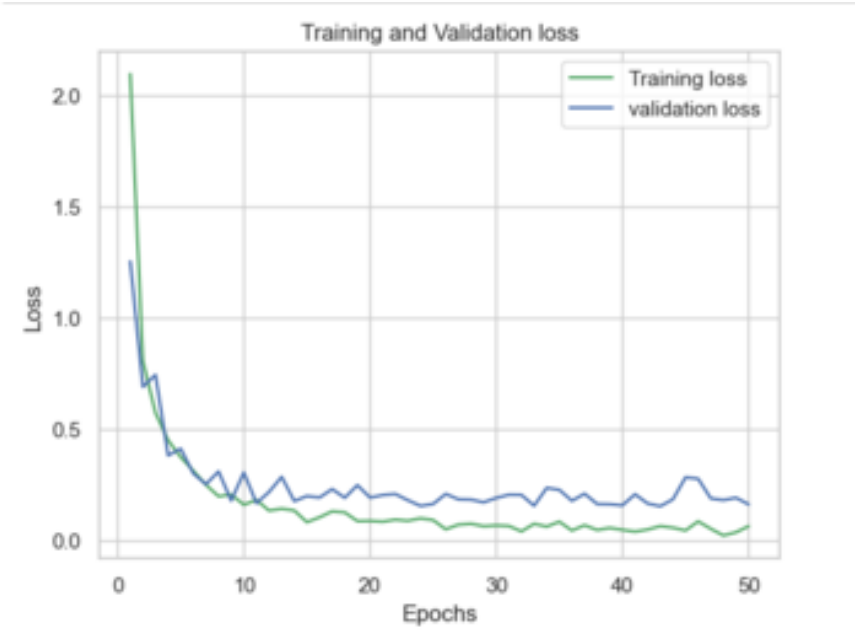


Figure 4.3: Average Loss curve.

The results unequivocally showcase in Table 4.1 the model's exceptional performance throughout the training period , spanning all epochs. The consistent decrease in loss over the 50 epochs signifies the model's effective generalization to non-visual data during the validation process. The classification report highlights the model's high performance for all categories, with high accuracy, recall, and f1 scores. Class 2 achieved the highest scores, with accuracy (0.99), recall (1.00), and f1 score (1.00), while the lowest scores were observed in the worst cases (accuracy: 0.83, recall: 0.78, f1 score: 0.81). Overall, the final results indicate a strong classification performance for the model, achieving an accuracy of 0.97.

Table 4.1: Classification Report of leaves Dataset.

	precision	recall	f1-score	support
0	1.00	0.97	0.99	502
1	1.00	0.98	0.99	373
2	0.99	1.00	1.00	517
3	1.00	0.99	1.00	545
4	1.00	1.00	1.00	520
5	0.99	1.00	1.00	528
6	0.99	0.99	0.99	549
7	1.00	1.00	1.00	556
8	0.98	1.00	0.99	519
9	0.98	1.00	0.99	524
10	0.83	0.78	0.81	212
11	0.81	0.83	0.82	187

12	0.97	0.78	0.87	184
13	0.0.87	0.95	0.91	167
14	0.86	0.87	0.86	193
15	0.80	0.91	0.85	178
accuracy			0.97	6254
macro avg	0.94	0.94	0.9	6254
weighted avg	0.97	0.97	0.97	6254

4.2.2 Product Dataset Results

Below are the results obtained from training the model on the dataset of products, which is categorized into 10 classes:

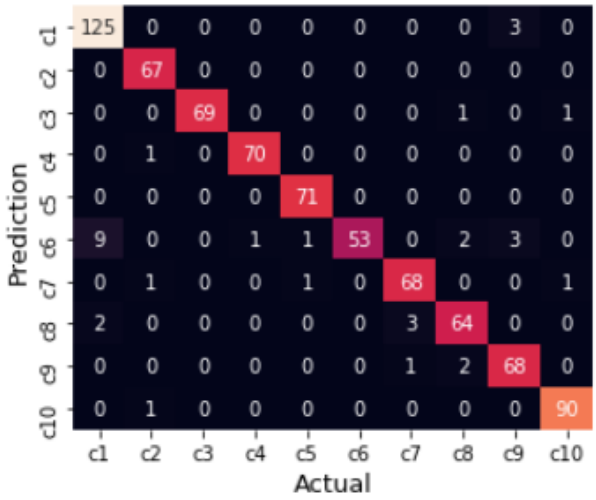


Figure 4.4: Confusion matrix of products Dataset.

Analysis of the Results The model achieved its best performance after 15 epochs of training on this dataset,as can be seen in the curves shown in Figures 4.5 and 4.6. Initially, the model had a loss value of 4.3632 and an accuracy of 0.2800. However, as the training progressed, the loss rate decreased to 0.0620, and the accuracy improved to 0.9831.

These results demonstrate that the model consistently provided highly accurate results throughout the 15 epochs. Notably, there was a significant improvement in both the loss value and accuracy, indicating that the model learned effectively during this training period.

Figure 4.5: Average Accuracy curve.

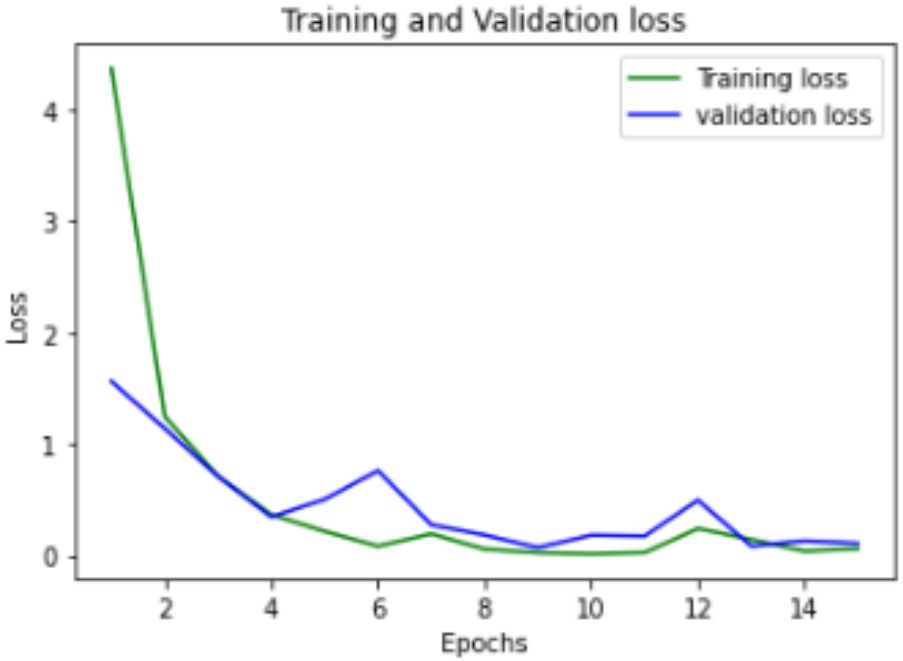


Figure 4.6: Average Loss curve.

Table 4.2: Classification Report of Product Dataset.

	precision	recall	f1-score	support
0	0.92	0.98	0.95	128
1	0.96	1.00	0.98	67
2	1.00	0.97	0.99	71
3	0.99	0.99	0.99	71
4	0.97	1.00	0.99	71
5	1.00	0.77	0.87	69
6	0.94	0.96	0.95	71
7	0.93	0.93	0.93	69
8	0.92	0.96	0.94	71
9	0.98	0.99	0.98	91
accuracy			0.96	779
macro avg	0.96	0.95	0.96	779
weighted avg	0.96	0.96	0.96	779

The provided table, referenced as Table 4.2, presents an analysis of the model's performance in classifying tomato produits dataset. It reveals notable accuracy, recall, and f1-scores across most categories, indicating the model's effectiveness. Class 2 stands out with a peak accuracy of 1.00, while class 5 exhibits lower results at 0.77, suggesting challenges and difficulties in accurately predicting this particular class. The weighted average scores for accuracy, recall, and f1-scores are calculated at 0.96, reflecting the model's strong classification capabilities on this dataset. Moreover, the model achieves an overall accuracy of 0.96, based on support from 779 instances.

Comparative Table and Data Analysis

Table 4.3: Comparing the Results of Several Dataset Training.

	Dataset				
	kaggle	Our	All	Product	Leaves
Train$_{A}cc$	0.9936	0.9996	0.9840	0.9831	0.9813
Train$_{L}oss$	0.0206	0.0011	0.570	0.0620	0.0634
Val$_{A}cc$	0.9975	0.9329	0.8859	0.9564	0.9677
Val$_{L}oss$	0.0099	0.4080	0.6557	0.1099	0.1615
Precision	1.00	0.94	0.90	0.96	0.97
Recall	1.00	0.93	0.89	0.96	0.97
F1$_{s}core$	1.00	0.93	0.89	0.96	0.97

There are two options for inputting data to the model when it comes to leaves: either including data with products or separating them. Table 4.3 demonstrates that separating the data yields better results

compared to combining them. This is because the model is specifically trained to identify leaf shapes, and introducing the variable of product shapes introduces ambiguity and potential errors in predictions.

4.3 Model Complexity and Execution Time Analysis

The increasing number of filters and the gradual reduction of spatial dimensions in the convolutional deep neural network architecture allow the model to capture and learn complex features from the input data. However, it is important to note that the model's complexity not only enhances its ability to learn complex patterns but also affects training time and resource requirements.
Table 3.4 provides a detailed description of the architecture of the convolutional deep neural network, showcasing the different layers and their configurations. This table serves as a valuable reference for understanding the model's structure, including the input and output shapes of each layer.

Figure 3.7 illustrates the visualization of the layer connections, offering a visual representation of the flow of information within the network. This helps to comprehend the hierarchy and connectivity between the layers, providing insights into the model's complexity.

Additionally, Table 3.5 presents the specific parameters of the model, including the number of filters, kernel size, and input/output dimensions. These parameters play a crucial role in determining the computational complexity and memory requirements of the model. By analyzing the parameter counts in each layer, we can estimate the memory requirements of the model. With a total of 1,752,048 parameters, this represents the memory necessary to store the model's weights and biases. Taking these factors into account allows for an accurate estimation of memory requirements, facilitating effective memory management and model optimization strategies.

The execution time of the model is influenced by the size of the input data and the complexity of the model architecture. When training the model with a bigger dataset of 19,161 images in the leaves dataset, the execution time is observed to be slower compared to processing a smaller number of images in the products dataset. The leaves dataset took approximately 29830 seconds to train, while the products dataset took approximately 872 seconds. This discrepancy in execution time can be attributed to the larger size of the leaves dataset, which requires more data to be processed, resulting in a longer computational time. Therefore, it is important to consider the dataset size when estimating and optimizing the execution time of the model.

Furthermore, the complexity of the model architecture also affects the execution time. Models with a higher number of layers or more complex structures generally require more time to process each input. As the model becomes more intricate, with an increased number of parameters and computations, the execution time is prolonged.

Taking these factors into account allows for a comprehensive understanding of the relationship between model complexity and execution time. By considering these factors and making informed decisions regarding optimization techniques, hardware choices, and resource allocation, researchers and practitioners can strike a balance between computational efficiency and model performance.

4.4 Visualization of The Application

To streamline the utilization of the model, a mobile application was developed, enabling users to effortlessly capture or import images from their phones for studying their respective statuses. The subsequent section elaborates on the range of services offered by the application.

4.4.1 Welcome Page

The application begins with a welcome interface that enables the user to choose the appropriate language, Arabic or English.

Figure 4.7: Welcome Page.

4.4.2 Login Page

This page allows you to log in to use the application by entering the information represented by the username and password.

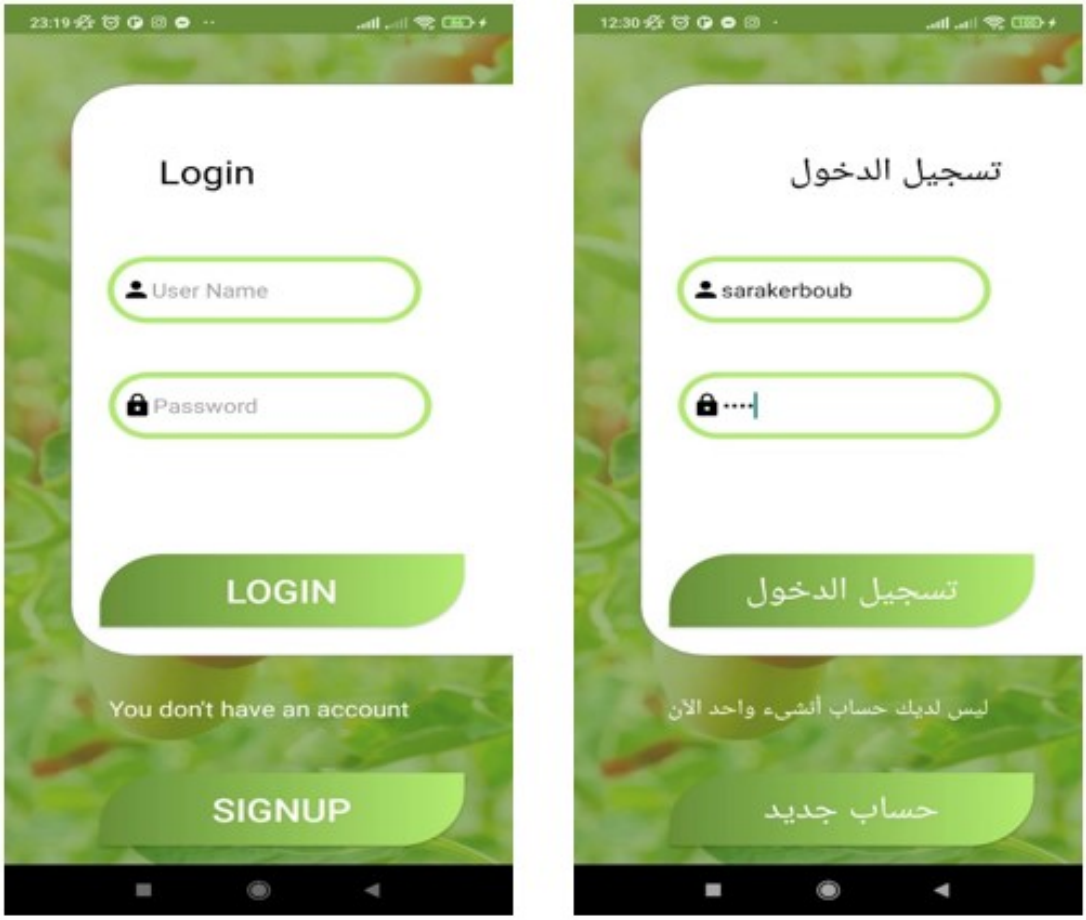


Figure 4.8: Sign in Page.

4.4.3 Sign Up Page

You must first obtain an account in the application, which requires the following information: username, email, password, phone number, and farmer ID card with the choice of the capacity as either a farmer or an expert, through the signup interface.

Figure 4.9: Sign Up page.

The Firebase database stores the information in this particular way.

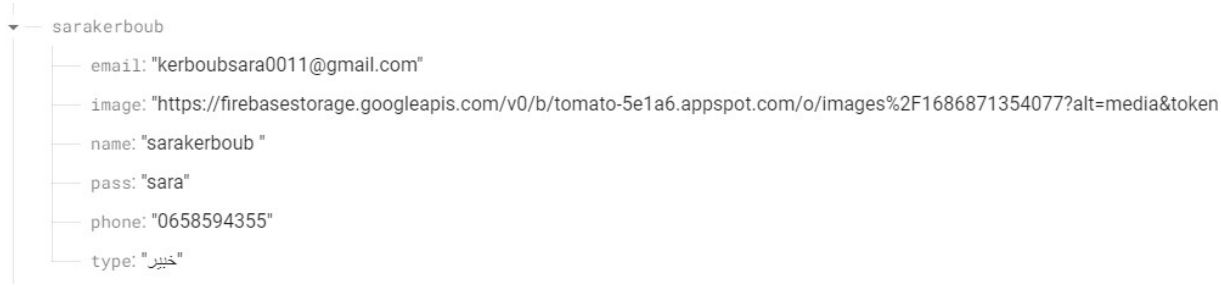


Figure 4.10: Sign Up page.

4.4.4 Home Page

The home interface in the application contains the main services represented in the detection of diseases in tomato leaves and products, also sending an email containing a video because it is a non-free service, as well as a group chat to share ideas between users.

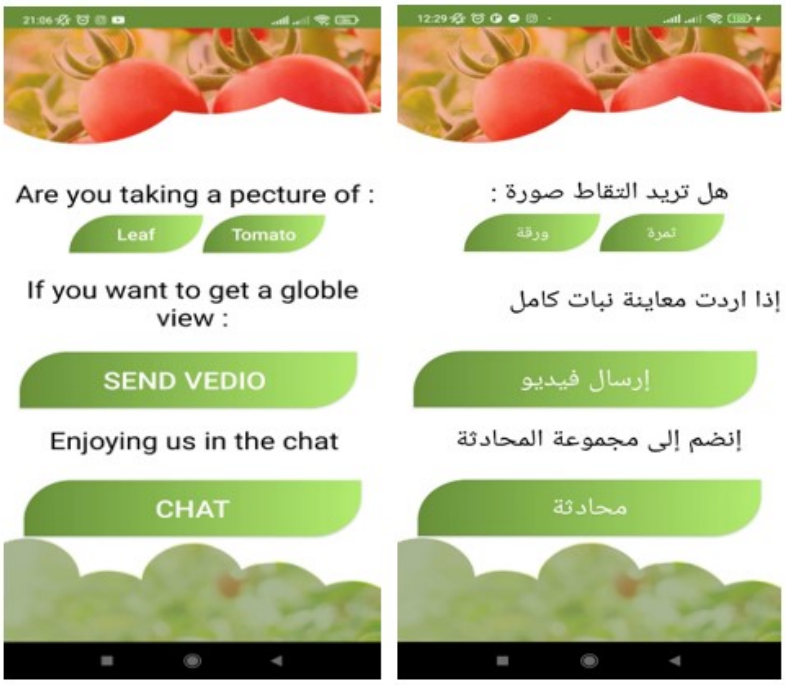


Figure 4.11: Home page.

4.4.4.1 Leaf and Tomato Button

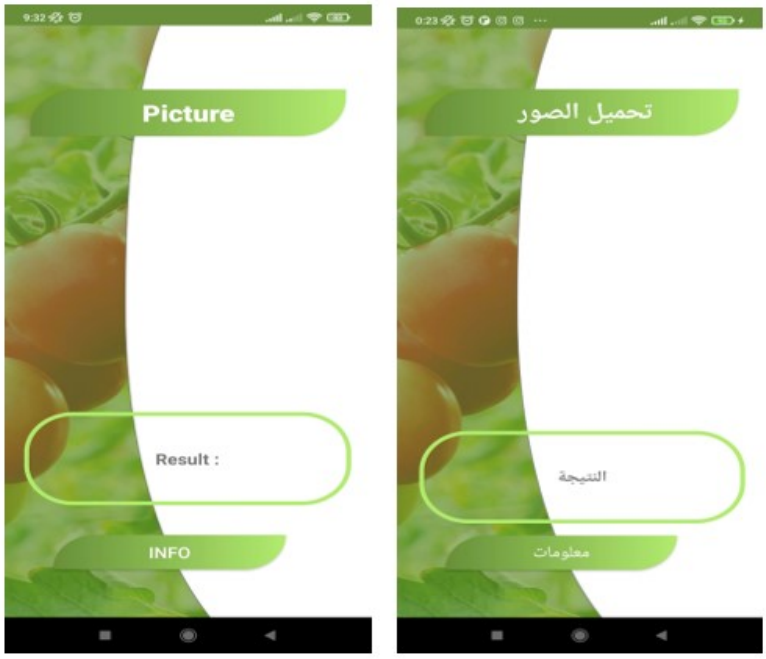


Figure 4.12: Import the image to Diagnose it Disease.

4.4.4.1.1 Picture Button

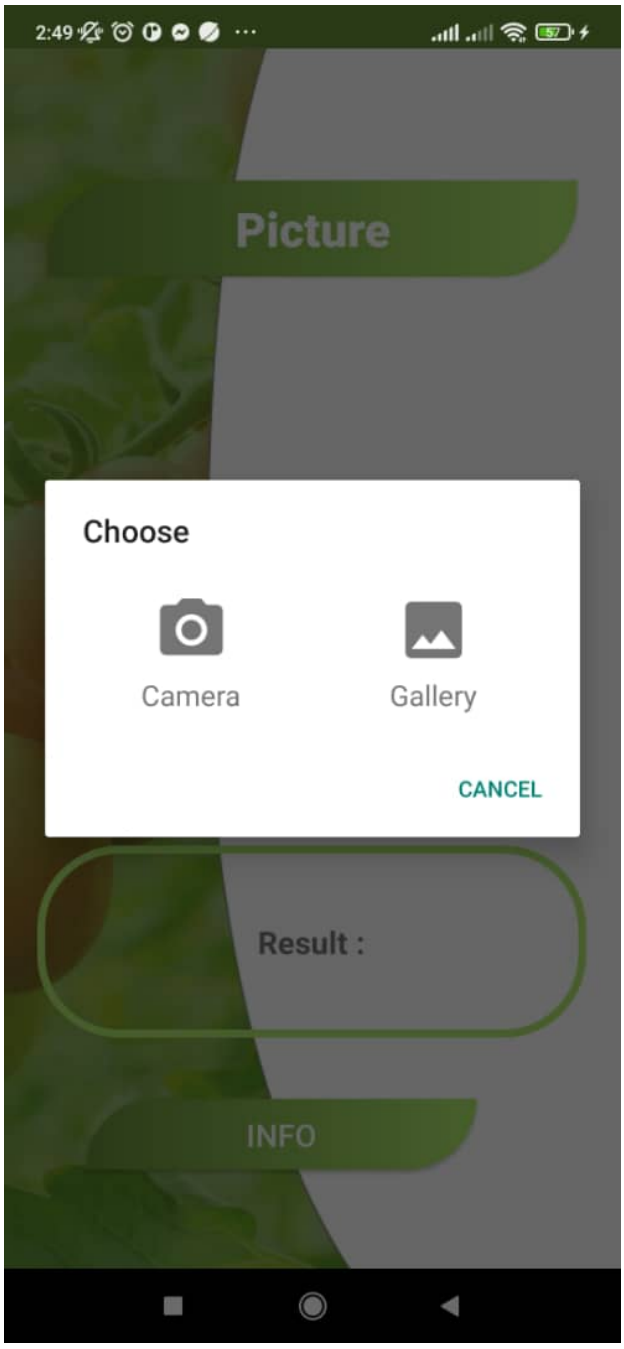


Figure 4.13: Image pick page.

4.4.4.2 Send Video Button

This interface makes it easy for the user to send a video via email of a complete plant for a report on the overall status.

Figure 4.14: Send Video to Diagnose its Disease..

4.4.4.2.1 Diagnostic Process Through Video

Once the video is received via email, the next step involves splitting it into individual frames. At this stage, a user interface is presented, enabling the selection of the desired video for the frame extraction process. The interface facilitates uploading of the chosen video from our device for the division, Once the task is completed, the execution is halted by pressing the button "Done".

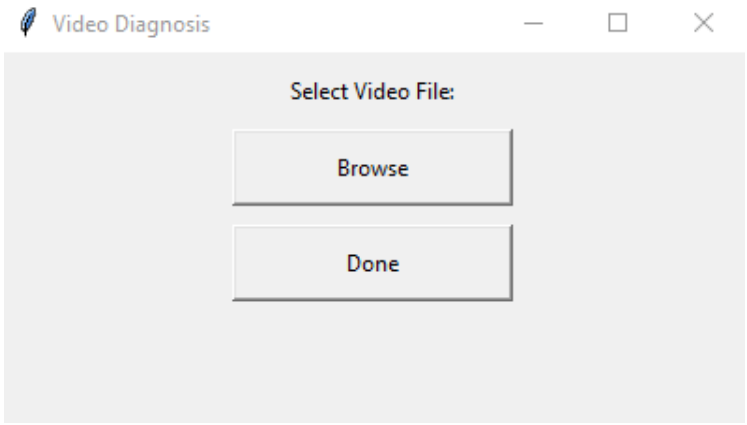


Figure 4.15: Spliting video interface.

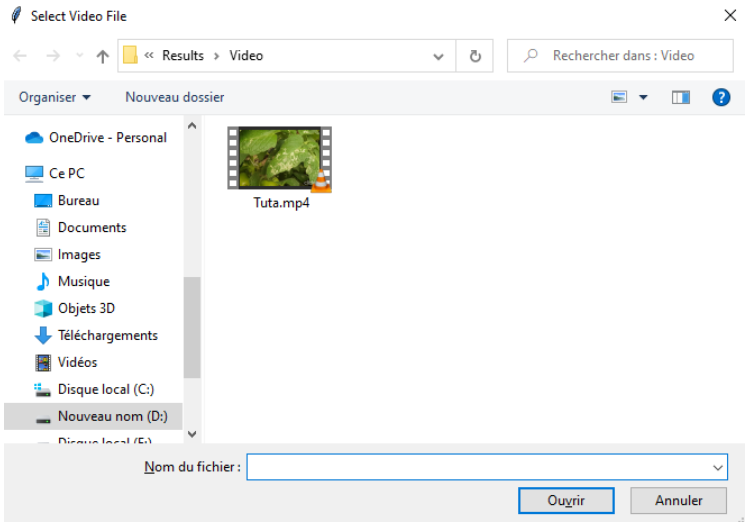


Figure 4.16: Select the sent video.

During this process, two files are created. The first file "Frames" stores frames as it captures a frame every half second to ensure comprehensive coverage of all angles. The second file "Images" stores split frames, separating them into multiple sections for easy access and organization.

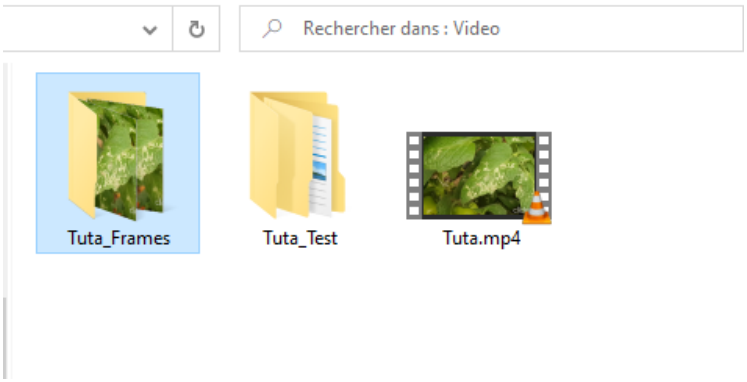


Figure 4.17: created files.

Once the division into sections is complete, each section undergoes a diagnostic process using the implemented CNN model. The resulting outcome is presented in the following format.

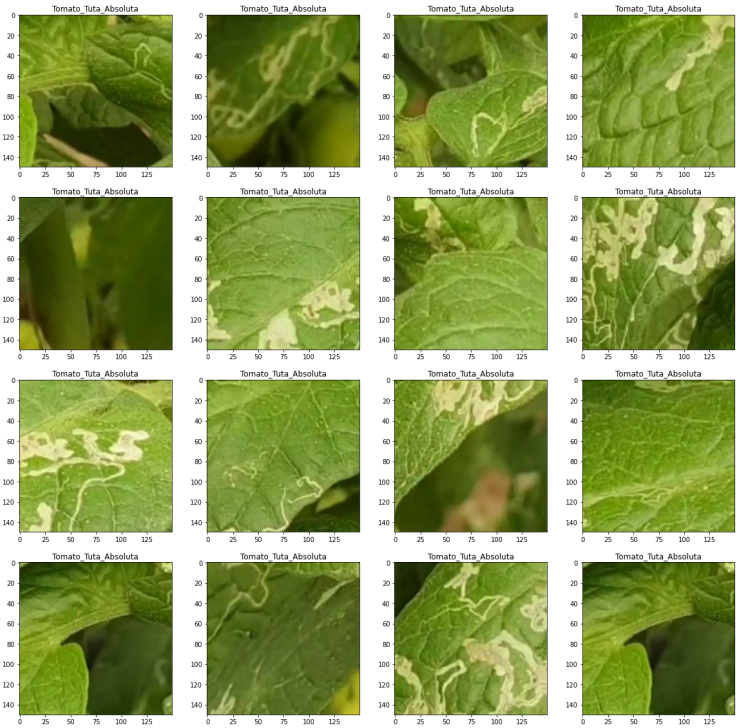


Figure 4.18: Classification phase .

4.4.4.3 Chat Room Button

The conversational interface allows users to exchange ideas, benefit from various experiences, and learn many ways to grow tomatoes.

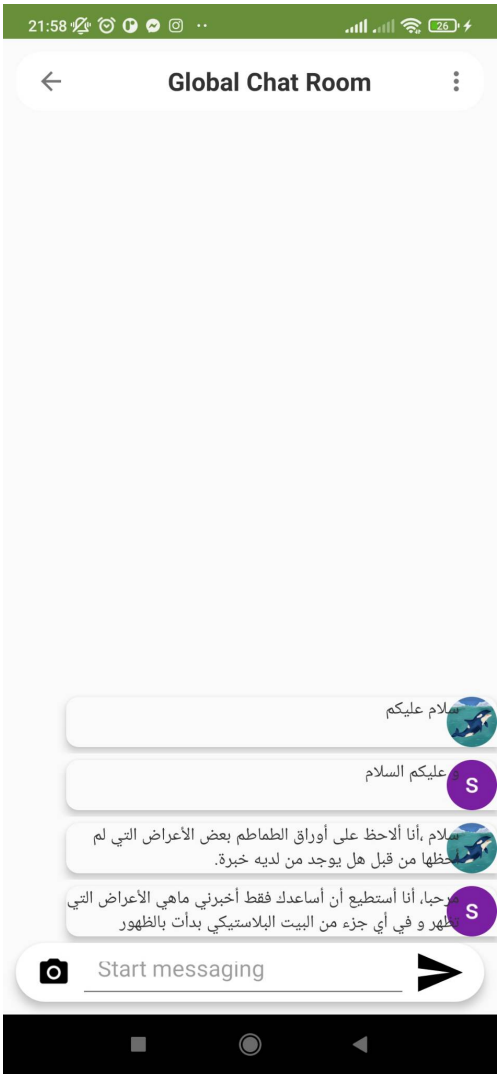


Figure 4.19: Chat Room page.

4.4.5 List Page

The settings interface shows user information and how to change the user name and password and the last function which asks for ratting the application.

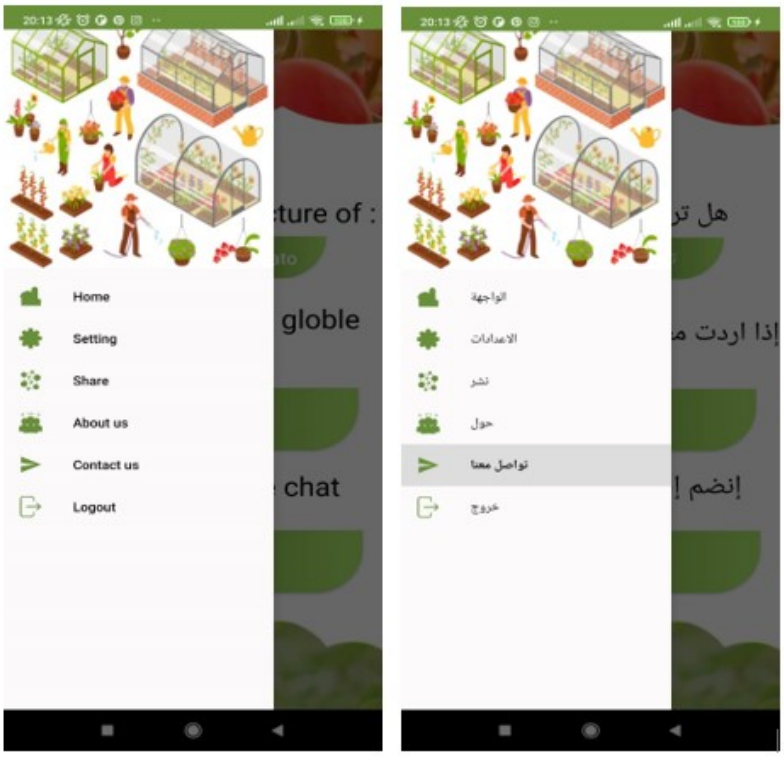


Figure 4.20: List Page.

4.4.5.1 Setting Button

This page is for the user account, where the account information is represented by the user name, email, and phone number, in addition to the account picture.

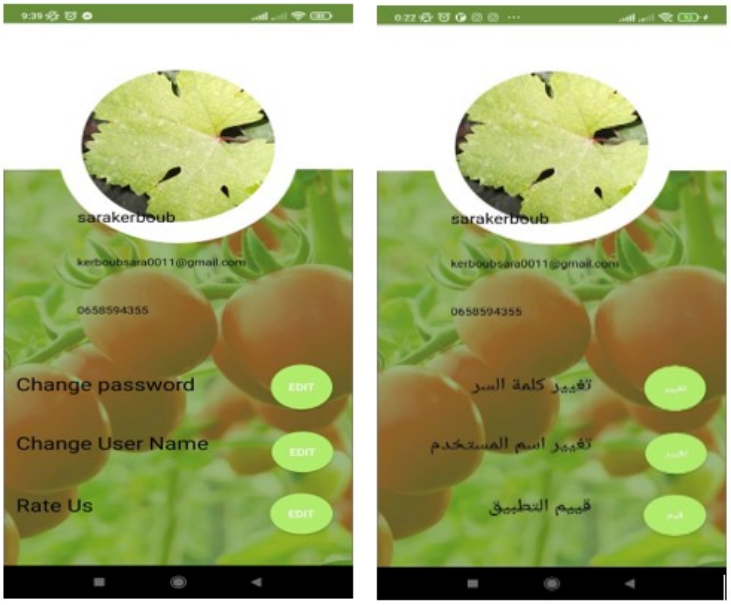


Figure 4.21: Setting Page.

4.4.5.1.1 Change Password Button

In case the user wants to change the user password, this interface provides this service.

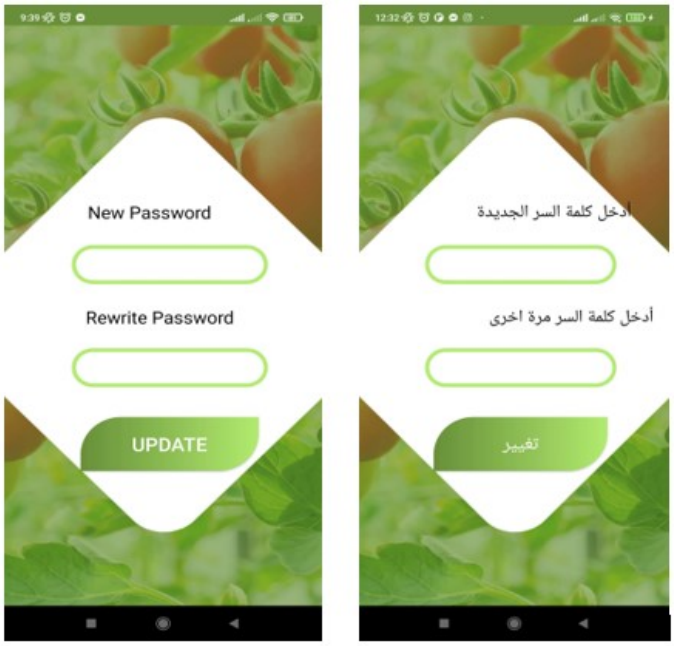


Figure 4.22: Change Password page.

4.4.5.1.2 Change User Name Button.

Users can change their account name by using this activity.

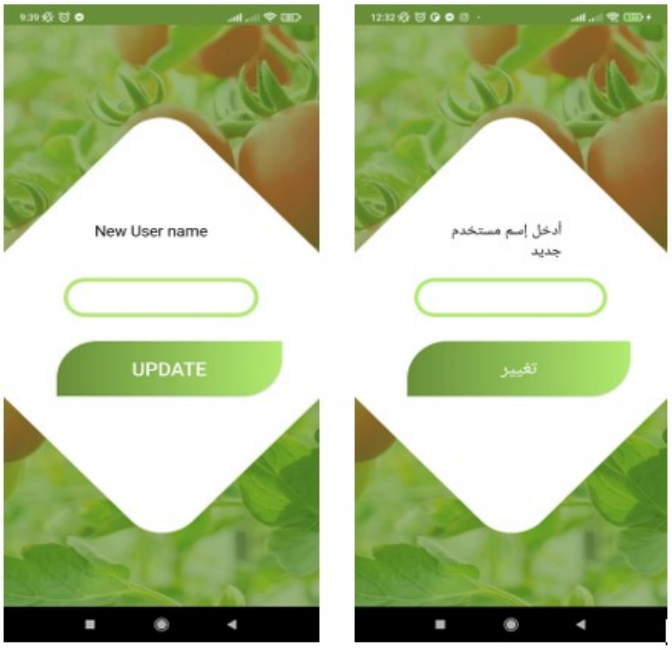


Figure 4.23: Change User Name page.

4.4.5.1.3 Rate Us Button.

We can see how much users like the application by submitting their evaluation using this dialog alert.

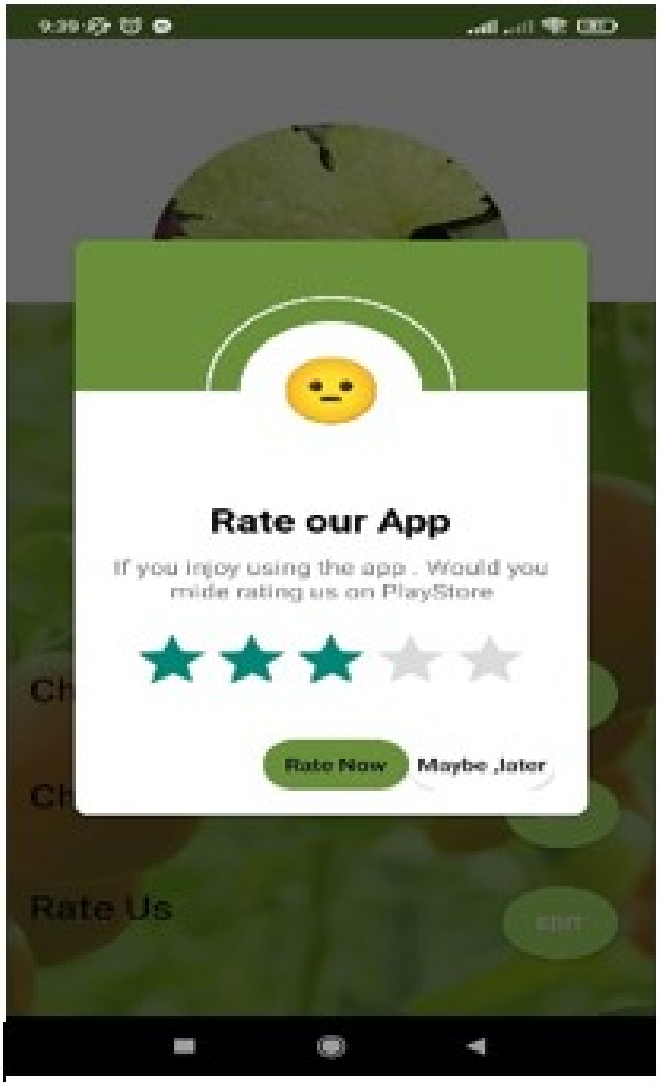


Figure 4.24: Rate Us page.

4.4.5.2 Share Button

In order to spread the application through friends in all social media, we suggested providing this interface to help publicize the application.

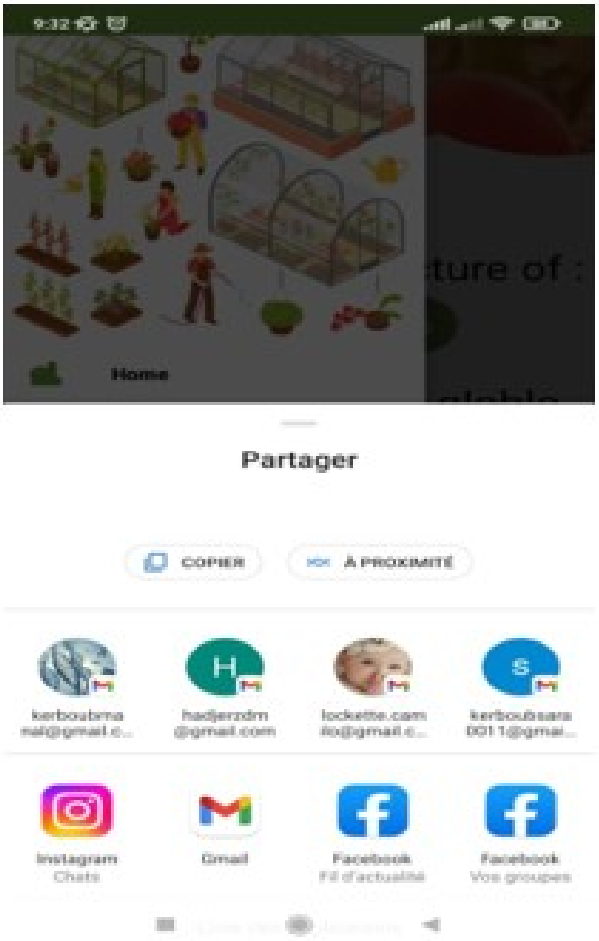


Figure 4.25: Share Page.

4.4.5.3 About us Button

It is necessary to provide an overview of the application and the services it provides to customers, in addition to the logo and the name of the application.

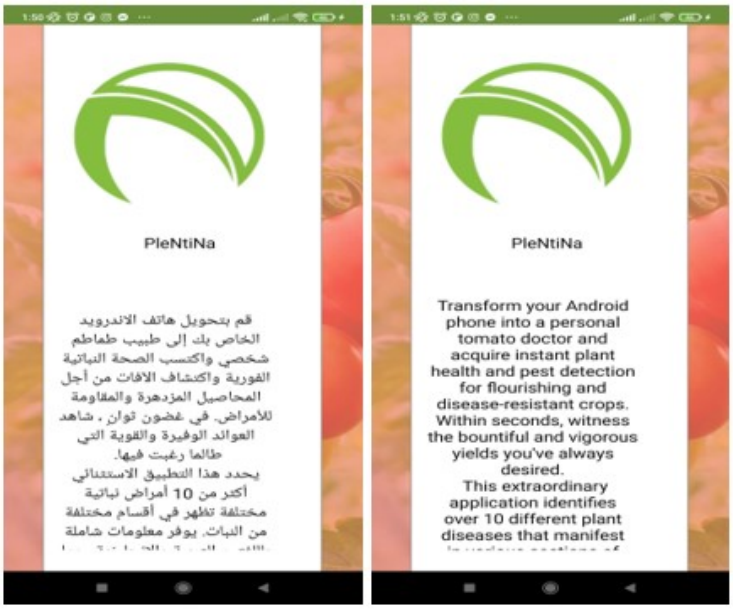


Figure 4.26: About us Page.

4.4.5.4 Contact us Button

We allow the user to express his needs and questions to us by sending the obvious e-mail from this page.

Figure 4.27: Contact us page.

4.4.5.5 Logout Button

The process of exiting the account is done through this dialog alert.

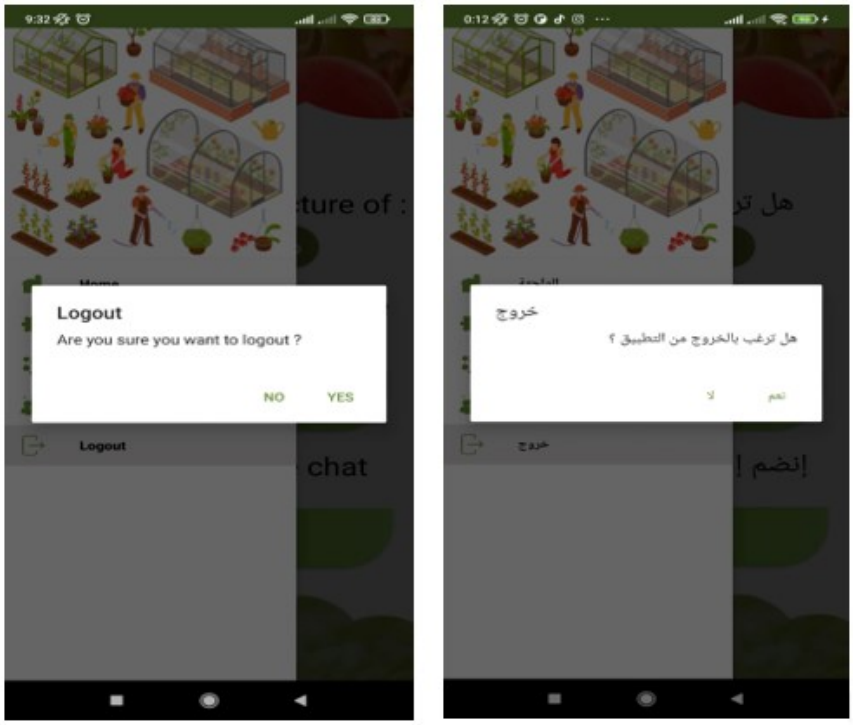


Figure 4.28: Logout page.

4.4.6 Leaf Detection Diseases Page

After clicking on the leaf button the application shows the interface responsible for importing leaves images (camera/gallery) to get the diagnosis of the diseases.

Figure 4.29: Leaf Detection Diseases Results.

4.4.7 Tomato Product Detection Diseases Page

In another way, the button tomato makes the user import a photo of a tomato product to know the disease.

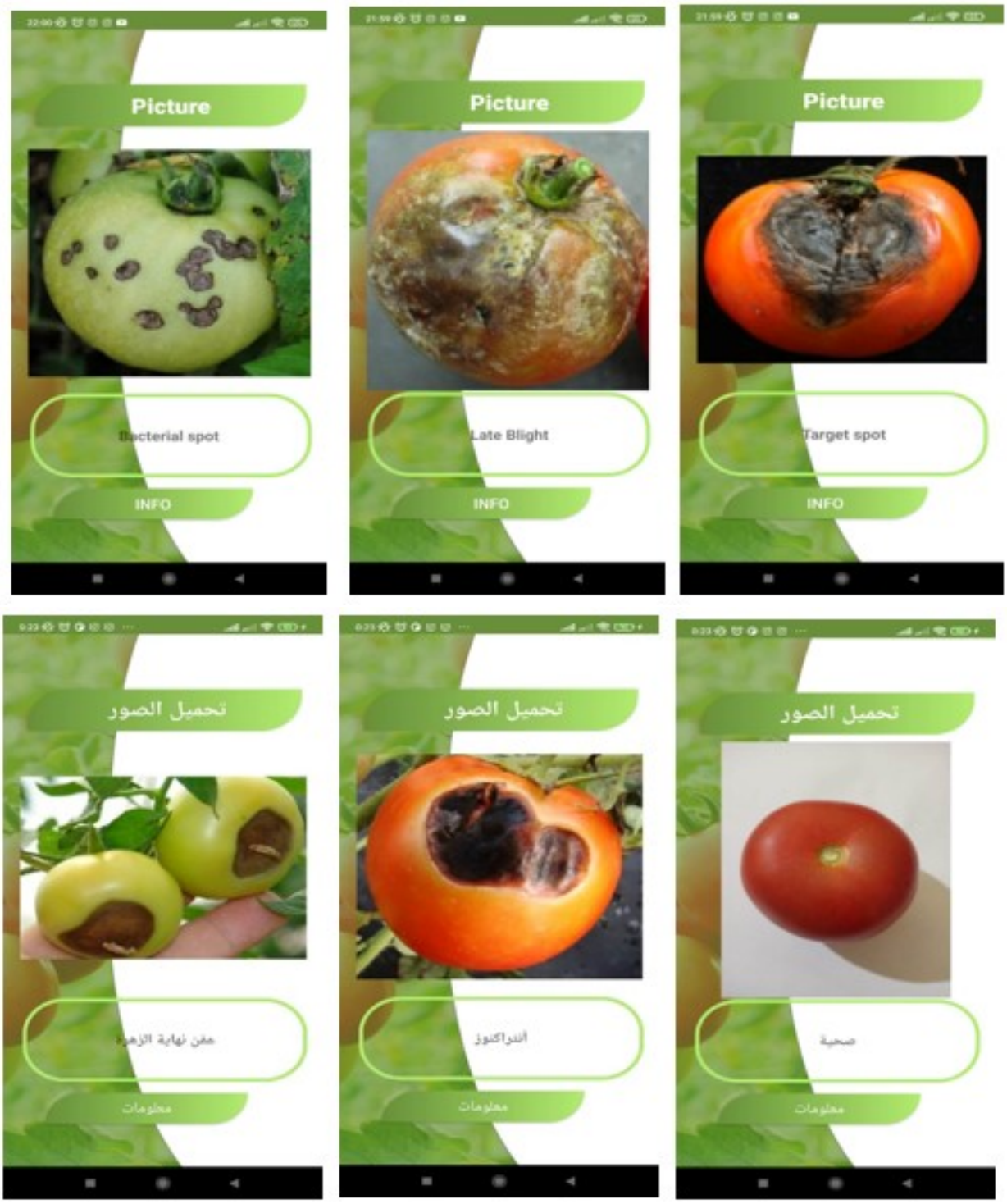


Figure 4.30: Product Detection Diseases Results..

4.4.8 Diseases Information Page

After identifying the disease appearing on the plant, this interface explains the condition, the reasons for its appearance, symptoms, and methods of treatment and protection.

Figure 4.31: Diseases Information Page.

4.5 Conclusion

This chapter shows all the results obtained in this work, where the results are divided into two parts. The first shows the results of the model that was tested on each of the leaves and fruits of the tomato plant. The second part was concerned with showing the application that was developed in order to facilitate the process of using the developed model. All interfaces that appear to the user while using the application are displayed.

General Conclusion

In this work, we have introduced a CNN-based model which is one of the deep learning techniques that treated images and extract features, the model was used to quantify tomato disseminated diseases. We directed our work and focus to the most prevalent diseases in the state of Biskra, where we supported the dataset with many pictures of these diseases, and we processed the data so that the model could be trained to give accurate and comprehensive results for all cases, in order to provide the farmer with the necessary treatment that he must follow to protect the crop.

Monitoring the growth of crops, especially tomatoes, and detecting diseases and pests is an integral part of the's life cycle. Additional nutrient treatment for the plant continues throughout. growing season and, if necessary, Determination of the type and amount of chemical needed, as well as the timing of spraying, based on regular monitoring of plant growth. Crop growth and disease and pest detection and disease can be monitored using interconnected IoT sensors that are connected to the internet[147]. Real-time data collection and decision-making by drones will play a critical role in accurately detecting diseases and pests. Drones have been used in precision agriculture because of their versatility they are capable of capturing image data used to detect infected areas and automatically spray the chemicals. useful-based detection[148].

As mentioned in each of [148],[147], and [149], a study has been done on how to integrate drones to aid in the identification of plant diseases, by determining the appropriate weight and shape of the drone. for the convenience of moving inside the greenhouse, this is one of the most critical difficulties, in addition to choosing the appropriate navigation algorithm.

SCIENCE CITE

International Conference On Health Information Technology And Future Prospects (ICHITFP-23)

27th Apr 2023, Constantine, Algeria

Paper Title: **Diagnosis Tomato's diseases based on Deep learning CNN**

Authors: **Barkat Selsabil, Kerboub Sara, Slatnia Sihem, Kazar Okba**

Paper ID: ***sciencecite_89635***

We are happy to inform you that your paper has been selected for (ICHITFP-23) on 27th Apr 2023, Constantine, Algeria after peer review process which will be organized by Science Cite, for presentation (Oral/poster Presentation) at the Conference. Registered papers will get Conference Proceeding having ISBN (International Standard Book Number) and certificates of paper presentation.

Kindly confirm your Registration and Event Participation by following links.

Official Page of event: https://www.sciencecite.com/event/index.php?id=1990381
For Registration guidelines: https://www.sciencecite.com/event/reg.php?id=1990381
Last date of registration: 20th April 2023

For any query related to payment you can mail us to- team@sciencecite.com

Note: Kindly send us the details regarding payment and Registration form to the official mail Id of the Event before last date of registration.

All the accepted scientific articles will be index in one of the following scientific metadata repository

Scopus® UNIVERSIDAD POLITÉCNICA PUERTO RICO ResearchPEDIA .org ScienceDirect

All Selected and registered papers will also be forwarded for publication in any one of the following International Journals after the conference.

Journal Name	Indexing and ISSN
International Journal of Research and Analytical Reviews (IJRAR)	Indexing : UGC ISSN: 2348-1269/2349-5138
Journal of Emerging Technologies and Innovative Research (JETIR)	Indexing : UGC ISSN: 2349-5162
International Journal of English Language, Literature in Humanities (IJELLH)	Indexing : UGC ISSN: 2321-7065
International Journal of Innovative Technology and Exploring Engineering (IJITEE)	Indexing : SCOPUS ISSN: 2278-3075/2278-3076
International Journal of Recent Technology and Engineering (IJRTE)	Indexing : SCOPUS ISSN: 2277-3878
Journal of Pure and Applied Microbiology (JPAM)	Indexing : SCOPUS ISSN: 0973-7510
International Journal of Supply Chain Management	Indexing : SCOPUS ISSN: 2050-7399/2051-3771
IOP Conference Series: Materials Science and Engineering	Indexing : SCOPUS ISSN: 1757-899X/1757-8981
International Journal of Project Organisation and Management	Indexing : SCOPUS ISSN: 1740-2905/1740-2891

Note:

This is the official letter for your paper Acceptance only. You need to confirm your registration and event participation by paying the registration fees before the last date of Registration.

Regards,

Akash Shinde
Convener
Science Cite
Contact: +91 9344535349
Email : team@sciencecite.com

Bibliography

[1] Xianlin Wang, Yuqing Liu, and Haohui Xin. Bond strength prediction of concrete-encased steel structures using hybrid machine learning method. *Structures*, 32:2279–2292, 09 2021.

[2] Ma Yan, Kang Liu, Zhibin Guan, Xu Xinkai, Xu Qian, and Hong Bao. Background augmentation generative adversarial networks (bagans): Effective data generation based on gan-augmented 3d synthesizing. *Symmetry*, 10:734, 12 2018.

[3] Shagan Sah. Machine learning: a review of learning types. 2020.

[4] Guangchun Ruan, Haiwang Zhong, Guanglun Zhang, Yiliu He, Xuan Wang, and Tianjiao Pu. Review of learning-assisted power system optimization, 06 2020.

[5] Xianlin Wang, Yuqing Liu, and Haohui Xin. Bond strength prediction of concrete-encased steel structures using hybrid machine learning method. *Structures*, 32:2279–2292, 09 2021.

[6] Danang Adi Pratama, Maharani Abu Bakar, Mustafa Man, and M Mashuri. Anns-based method for solving partial differential equations: a survey. 2021.

[7] M Kumar, NS Raghuwanshi, and R Singh. Artificial neural networks approach in evapotranspiration modeling: a review. *Irrigation science*, 29:11–25, 2011.

[8] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. *Towards Data Sci*, 6(12):310–316, 2017.

[9] Yongmei Ren, Jie Yang, Qingnian Zhang, and Zhiqiang Guo. Multi-feature fusion with convolutional neural network for ship classification in optical images. *Applied Sciences*, 9(20):4209, 2019.

[10] Koustav Dutta, Rasmita Lenka, and Selim Sarowar. Improvement of denoising in images using generic image denoising network (gid net). 02 2022.

[11] Hossein Gholamalinejad and Hossein Khosravi. Pooling methods in deep neural networks, a review, 09 2020.

[12] Charlotte Pelletier, Geoffrey Webb, and François Petitjean. Temporal convolutional neural network for the classification of satellite image time series. *Remote Sensing*, 11:523, 03 2019.

[13] Yiren Zhou, Sibo Song, and Ngai-Man Cheung. On classification of distorted images with deep convolutional neural networks. 01 2017.

[14] Danang Adi Pratama, Maharani Abu Bakar, Mustafa Man, and M Mashuri. Anns-based method for solving partial differential equations: a survey. 2021.

[15] Usman Muhammad, Weiqiang Wang, Shahbaz Pervaiz Chattha, and Sajid Ali. Pre-trained vggnet architecture for remote-sensing image scene classification. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 1622–1627. IEEE, 2018.

[16] Taqdeer Gill, Simranveer Gill, Yuvraj Chopra, Jason Koff, Dinesh Saini, and Karansher Sandhu. A comprehensive review of high throughput phenotyping and machine learning for plant stress phenotyping. *Plant Phenomics*, 2:3, 04 2022.

[17] Manoj Kumar, Maharishi Tomar, Deep Jyoti Bhuyan, Sneh Punia, Simona Grasso, Amanda Gomes Almeida Sa, Bruno Augusto Mattar Carciofi, Fatima Arrutia, Sushil Changan, Surinder Singh, et al. Tomato (solanum lycopersicum l.) seed: A review on bioactives and biomedical activities. *Biomedicine & Pharmacotherapy*, 142:112018, 2021.

[18] Redmond Ramin Shamshiri, James W Jones, Kelly R Thorp, Desa Ahmad, Hasfalina Che Man, and Sima Taheri. Review of optimum temperature, humidity, and vapour pressure deficit for microclimate evaluation and control in greenhouse cultivation of tomato: a review. *International agrophysics*, 32(2):287–302, 2018.

[19] Dominique Blancard. *Tomato diseases: identification, biology and control: a colour handbook*. CRC Press, 2012.

[20] Sandro Augusto Magalhães, Luís Castro, Germano Moreira, Filipe Neves Dos Santos, Mário Cunha, Jorge Dias, and António Paulo Moreira. Evaluating the single-shot multibox detector and yolo deep learning models for the detection of tomatoes in a greenhouse. *Sensors*, 21(10):3569, 2021.

[21] Andrea Giovanni Caruso, Sofia Bertacca, Giuseppe Parrella, Roberto Rizzo, Salvatore Davino, and Stefano Panno. Tomato brown rugose fruit virus: A pathogen that is changing the tomato production worldwide. *Annals of Applied Biology*, 181(3):258–274, 2022.

[22] Bilan annuel des statistiques de la ministère de l'agriculture, du développement rural et de la pêche, 2016.

[23] CD Jones, JB Jones, and WS Lee. Diagnosis of bacterial spot of tomato using spectral signatures. *Computers and Electronics in Agriculture*, 74(2):329–335, 2010.

[24] Tomato leaf disease detection. https://www.kaggle.com/datasets/kaustubhb999/tomatoleaf.

[25] Frank Krüger. *Activity, Context, and Plan Recognition with Computational Causal Behaviour Models*. PhD thesis, 12 2016.

[26] Renata Galhardo Borguini and Elizabeth A Ferraz da Silva Torres. Tomatoes and tomato products as dietary sources of antioxidants. *Food Reviews International*, 25(4):313–325, 2009.

[27] Ibtisam Zoaibi and Obira Donia. A study on greenhouses, pros and cons. Master's Graduation Note, 2021.

[28] Qimei Wang, Feng Qi, Minghe Sun, Jianhua Qu, Jie Xue, et al. Identification of tomato disease types and detection of infected areas based on deep convolutional neural networks and object detection techniques. *Computational intelligence and neuroscience*, 2019, 2019.

[29] Djida Ayad, Djihad Aribi, Bruno Hamon, Abdelaziz Kedad, Philippe Simoneau, and Zouaoui Bouznad. Distribution of large-spored alternaria species associated with early blight of potato and tomato in algeria. *Phytopathologia Mediterranea*, 58(1):139–150, 2019.

[30] Stavan Ruparelia, Monil Jethva, and Ruchi Gajjar. Real-time tomato detection, classification, and counting system using deep learning and embedded systems. In *Proceedings of the International e-Conference on Intelligent Systems and Signal Processing: e-ISSP 2020*, pages 511–522. Springer, 2022.

[31] Mengping Dong, Shaomin Mu, Tingting Su, and Wenjie Sun. Image recognition of peanut leaf diseases based on capsule networks. In *Artificial Intelligence: Second CCF International Conference, ICAI 2019, Xuzhou, China, August 22-23, 2019, Proceedings 2*, pages 43–52. Springer, 2019.

[32] Dulari Bhatt, Chirag Patel, Hardik Talsania, Jigar Patel, Rasmika Vaghela, Sharnil Pandya, Kirit Modi, and Hemant Ghayvat. Cnn variants for computer vision: history, architecture, application, challenges and future scope. *Electronics*, 10(20):2470, 2021.

[33] Belal AM Ashqar and Samy S Abu-Naser. Image-based tomato leaves diseases detection using deep learning. 2018.

[34] Mohammad Mustafa Taye. Understanding of machine learning with deep learning: Architectures, workflow, applications and future directions. *Computers*, 12(5), 2023.

[35] Dulari Bhatt, Chirag Patel, Hardik Talsania, Jigar Patel, Rasmika Vaghela, Sharnil Pandya, Kirit Modi, and Hemant Ghayvat. Cnn variants for computer vision: History, architecture, application, challenges and future scope. *Electronics*, 10(20), 2021.

[36] Danny Weyns. Software engineering of self-adaptive systems. *Handbook of software engineering*, pages 399–443, 2019.

[37] Batta Mahesh. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, 9:381–386, 2020.

[38] Taiwo Oladipupo Ayodele. Types of machine learning algorithms. *New advances in machine learning*, 3:19–48, 2010.

[39] Mahamed Omran, Andries Engelbrecht, and Ayed Salman. An overview of clustering methods. *Intell. Data Anal.*, 11:583–605, 11 2007.

[40] Iqbal H Sarker. Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, 2(3):160, 2021.

[41] Matthew Botvinick, Jane X Wang, Will Dabney, Kevin J Miller, and Zeb Kurth-Nelson. Deep reinforcement learning and its neuroscientific implications. *Neuron*, 107(4):603–616, 2020.

[42] Jinming Zou, Yi Han, and Sung-Sau So. Overview of artificial neural networks. *Artificial neural networks: methods and applications*, pages 14–22, 2009.

[43] Peng Zhang and Yuval Itan. Biological network approaches and applications in rare disease studies. *Genes*, 10(10):797, 2019.

[44] Carlos Gershenson. Artificial neural networks for beginners. *arXiv preprint cs/0308031*, 2003.

[45] Mohammad-Parsa Hosseini, Amin Hosseini, and Kiarash Ahi. A review on machine learning for eeg signal processing in bioengineering. *IEEE reviews in biomedical engineering*, 14:204–218, 2020.

[46] Claude Touzet. *LES RESEAUX DE NEURONES ARTIFICIELS, INTRODUCTION AU CONNEXIONNISME : COURS, EXERCICES ET TRAVAUX PRATIQUES*, volume 129 of *EC2*. 1992.

[47] Christian W Dawson and Robert Wilby. An artificial neural network approach to rainfall-runoff modelling. *Hydrological Sciences Journal*, 43(1):47–66, 1998.

[48] Ivan Nunes Da Silva, Danilo Hernane Spatti, Rogerio Andrade Flauzino, Luisa Helena Bartocci Liboni, Silas Franco dos Reis Alves, Ivan Nunes da Silva, Danilo Hernane Spatti, Rogerio Andrade Flauzino, Luisa Helena Bartocci Liboni, and Silas Franco dos Reis Alves. *Artificial neural network architectures and training processes*. Springer, 2017.

[49] Mohammed Imran and Sarah A Alsuhaibani. A neuro-fuzzy inference model for diabetic retinopathy classification. In *Intelligent Data Analysis for Biomedical Applications*, pages 147–172. Elsevier, 2019.

[50] Jianli Feng and Shengnan Lu. Performance analysis of various activation functions in artificial neural networks. In *Journal of physics: conference series*, volume 1237, page 022030. IOP Publishing, 2019.

[51] Jun Han and Claudio Moraga. The influence of the sigmoid function parameters on the speed of backpropagation learning. In *From Natural to Artificial Neural Computation: International Workshop on Artificial Neural Networks Malaga-Torremolinos, Spain, June 7– 9, 1995 Proceedings 3*, pages 195–201. Springer, 1995.

[52] Mohit Sewak, Sanjay K Sahay, and Hemant Rathore. An overview of deep learning architecture of deep neural networks and autoencoders. *Journal of Computational and Theoretical Nanoscience*, 17(1):182–188, 2020.

[53] Kunal Banerjee, Rishi Raj Gupta, Karthik Vyas, Biswajit Mishra, et al. Exploring alternatives to softmax function. *arXiv preprint arXiv:2011.11538*, 2020.

[54] Pierre Blanchard, Desmond J Higham, and Nicholas J Higham. Accurate computation of the log-sum-exp and softmax functions. *arXiv preprint arXiv:1909.03469*, 2019.

[55] Pramila P Shinde and Seema Shah. A review of machine learning and deep learning applications. In *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*, pages 1–6. IEEE, 2018.

[56] Deepack Jakhar and Ishmeet Kaur. Artificial intelligence, machine learning and deep learning: definitions and differences. *Clinical and experimental dermatology*, 45(1):131–132, 2020.

[57] Robert DiPietro and Gregory D. Hager. Chapter 21 - deep learning: Rnns and lstm. In S. Kevin Zhou, Daniel Rueckert, and Gabor Fichtinger, editors, *Handbook of Medical Image Computing and Computer Assisted Intervention*, The Elsevier and MICCAI Society Book Series, pages 503–519. Academic Press, 2020.

[58] Tarun Kumar Gupta and Khalid Raza. Chapter 7 - optimization of ann architecture: A review on nature-inspired techniques. In Nilanjan Dey, Surekha Borra, Amira S. Ashour, and Fuqian Shi, editors, *Machine Learning in Bio-Signal Analysis and Diagnostic Imaging*, pages 159–182. Academic Press, 2019.

[59] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, pages 184–199. Springer, 2014.

[60] Ahmed Ali Mohammed Al-Saffar, Hai Tao, and Mohammed Ahmed Talab. Review of deep convolution neural network in image classification. In *2017 International conference on radar, antenna, microwave, electronics, and telecommunications (ICRAMET)*, pages 26–31. IEEE, 2017.

[61] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

[62] Dulari Bhatt, Chirag Patel, Hardik Talsania, Jigar Patel, Rasmika Vaghela, Sharnil Pandya, Kirit Modi, and Hemant Ghayvat. Cnn variants for computer vision: history, architecture, application, challenges and future scope. *Electronics*, 10(20):2470, 2021.

[63] Dulari Bhatt, Chirag Patel, Hardik Talsania, Jigar Patel, Rasmika Vaghela, Sharnil Pandya, Kirit Modi, and Hemant Ghayvat. Cnn variants for computer vision: history, architecture, application, challenges and future scope. *Electronics*, 10(20):2470, 2021.

[64] Saad Albawi and Osman Ucan. Social touch gesture recognition using convolutional neural network. *Computational Intelligence and Neuroscience*, 2018, 10 2018.

[65] Hossein Gholamalinezhad and Hossein Khosravi. Pooling methods in deep neural networks, a review. *arXiv preprint arXiv:2009.07485*, 2020.

[66] Afia Zafar, Muhammad Aamir, Nazri Mohd Nawi, Ali Arshad, Saman Riaz, Abdulrahman Alruban, Ashit Kumar Dutta, and Sultan Almotairi. A comparison of pooling methods for convolutional neural networks. *Applied Sciences*, 12(17):8643, 2022.

[67] Arohan Ajit, Koustav Acharya, and Abhishek Samanta. A review of convolutional neural networks. In *2020 international conference on emerging trends in information technology and engineering (ic-ETITE)*, pages 1–5. IEEE, 2020.

[68] Tianmei Guo, Jiwen Dong, Henjian Li, and Yunxing Gao. Simple convolutional neural network on image classification. pages 721–724, 2017.

[69] Xuefeng Liu, Qiaoqiao Sun, Yue Meng, Congcong Wang, and Min Fu. Feature extraction and classification of hyperspectral image based on 3d-convolution neural network. pages 918–922, 2018.

[70] Ibrahem Kandel and Mauro Castelli. How deeply to fine-tune a convolutional neural network: a case study using a histopathology dataset. *Applied Sciences*, 10(10):3359, 2020.

[71] Shadman Sakib, Nazib Ahmed, Ahmed Jawad Kabir, and Hridon Ahmed. An overview of convolutional neural network: its architecture and applications. 2019.

[72] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[73] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. 63(11), 2020.

[74] Hamed Alqahtani, Manolya Kavakli-Thorne, and Gulshan Kumar. Applications of generative adversarial networks (gans): an updated review. *Archives of Computational Methods in Engineering*, 28(2):525–552, March 2021.

[75] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[76] Wei Ying, Yu Zhang, Junzhou Huang, and Qiang Yang. Transfer learning via learning to transfer. In *International conference on machine learning*, pages 5085–5094. PMLR, 2018.

[77] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[78] Nsenge Mpia Héritier and Inipaivudu Baelani Nephtali. L'algorithme de rétro-propagation de gradient dans le perceptron multicouche: Bases et étude de cas. *International Journal of Innovation and Applied Studies*, 32(2):271–290, 2021.

[79] David R Harris and Dorian Q Fuller. Agriculture: definition and overview. *Encyclopedia of global archaeology*, pages 104–113, 2014.

[80] Sarah Velten, Julia Leventon, Nicolas Jager, and Jens Newig. What is sustainable agriculture? a systematic review. *Sustainability*, 7(6):7833–7865, 2015.

[81] Kaiming Wu, Mi Kyung Kim, Sangmi Park, and Yu-Jin Yang. Importance of dietary phytochemicals in plant-based diet. *Journal of medicinal food*, 17(7):707–718, 2014.

[82] Marcin Baranski, Dominika Srednicka-Tober, Nikolaos Volakakis, Chris Seal, Roy Sanderson, Gavin B Stewart, Charles Benbrook, Bruno Biavati, Emilia Markellou, Charilaos Giotis, et al. Higher antioxidant and lower cadmium concentrations and lower incidence of pesticide residues in organically grown crops: a systematic literature review and meta-analyses. *The British journal of nutrition*, 112(5):794–811, 2014.

[83] Juan Antonio Carrasco-González, Irma Osuna-Ruíz, María de Lourdes Peralta Reyes-Escogido, and Dulce María Barradas-Dermitz. Nutritional value and phytochemical components of organically and conventionally grown vegetables. *Journal of food science and technology*, 56(5):2465–2472, 2019.

[84] Robyn C Gilden, Katie Huffling, and Barbara Sattler. Pesticides and health risks. *Journal of Obstetric, Gynecologic, & Neonatal Nursing*, 39(1):103–110, 2010.

[85] Jiaojiao Zhu, Xiaodong Gao, Yujuan Wen, Xiaohui Liang, Qiuxiang Zhang, Xinping Chen, and Qirong Shen. High nitrogen concentration alters tomato leaf economics by decreasing leaf carotenoids and thickness. *Journal of Plant Physiology*, 232:200–209, 2019.

[86] Aneta Gerszberg, Katarzyna Hnatuszko-Konka, Tomasz Kowalczyk, and Andrzej K Kononowicz. Tomato (solanum lycopersicum l.) in the service of biotechnology. *Plant Cell, Tissue and Organ Culture (PCTOC)*, 120:881–902, 2015.

[87] J Benton Jones Jr. *Tomato plant culture: in the field, greenhouse, and home garden*. CRC press, 2007.

[88] Pardeep Kumar Duhan. Cost benefit analysis of tomato production in protected and open farm. *International Journal of Advanced Research in Management and Social Sciences*, 5(12):140–148, 2016.

[89] Cheiri Kubota, A de Gelder, and Mary M Peet. Greenhouse tomato production. In *Tomatoes*, pages 276–313. CABI Wallingford UK, 2018.

[90] J Benton Jones Jr. *Tomato plant culture: in the field, greenhouse, and home garden*. CRC press, 2007.

[91] Souraya BEN AICHI. Enquête sur la filière tomate dans la région des ziban comparaison entre deux systèmes de culture (le tunnel et le canarien). Master's thesis, Université Mohamed Khider de Biskra, Faculté des Sciences de la Nature et de la Vie, Sciences Agronomiques, 2019.

[92] ResearchAndMarkets. Tomato market - growth, trends, and forecasts (2020 - 2025). https://www.researchandmarkets.com/reports/5383368/tomato-market-growth-trends-and-forecasts, 2021.

[93] FAO. Faostat: Crops production. http://www.fao.org/faostat/en/#data/QC/visualize, 2021.

[94] USDA Foreign Agricultural Service. Tomato: World markets and trade. https://apps.fas.usda.gov/psdonline/circulars/tomato.pdf, 2021.

[95] Ministry of Agriculture and Rural Development. Algeria's agricultural sector in brief. http://www.minagri.dz/index.php/fr/actualites/la-direction-generale-des-politiques-et-des-strategies-agricoles/252-le-secteur-agricole-en-algerie-en-bref, 2019.

[96] International Trade Centre. Algeria: Market profile. https://www.trademap.org/Country_SelProduct.aspx?nvpm=1%7c012%7c%7c%7c%7c0702%7c%7c%7c1%7c1%7c2%7c2%7c2%7c1%7c1%7c1%7c1%7c1, 2021.

[97] United States Department of Agriculture. Algeria: Exporter guide. https://www.fas.usda.gov/data/algeria-exporter-guide-2021, 2021.

[98] Kerry F Pedley and Gregory B Martin. Molecular basis of pto-mediated resistance to bacterial speck disease in tomato. *Annual review of phytopathology*, 41(1):215–243, 2003.

[99] Sadikshya Sharma and Krishna Bhattarai. Progress in developing bacterial spot resistance in tomato. *Agronomy*, 9(1):26, 2019.

[100] Thomas A Zitter. Bacterial diseases of tomato. 1985.

[101] W Harry Lange and Lorin Bronson. Insect pests of tomatoes. *Annual review of entomology*, 26(1):345–371, 1981.

[102] D. Michael Benson. *Tomato Diseases: A Practical Guide for Seedsmen, Growers, and Agricultural Advisors*. American Phytopathological Society, St. Paul, MN, 2nd edition, 1993.

[103] Safia Chougar. *Bioécologie de la mineuse de la tomateTuta absoluta (Meyrick, 1917) dans la région de Tizi Ouzou. Esssais de lutte*. PhD thesis, Universite Mouloud MAMMERI Tizi-Ouzou, 2020.

[104] Roger AC Jones. Global plant virus disease pandemics and epidemics. *Plants*, 10(2):233, 2021.

[105] Mark Lawrence Gleason and Brooke A Edmunds. *Tomato diseases and disorders*. Iowa State University, University Extension Ames, IA, 2005.

[106] Thomas A Zitter and R Provvidenti. Virus diseases and disorders of tomato. 1984.

[107] T Kalb. Tomato diseases: tomato mosaic virus (tomv). Technical report, AVRDC-The World Vegetable Center, 2004.

[108] Anirudh Srinivasan Chakravarthy and Sundaresan Raman. Early blight identification in tomato leaves using deep learning. In *2020 International conference on contemporary computing and applications (IC3A)*, pages 154–158. IEEE, 2020.

[109] Mark Lawrence Gleason and Brooke Aurora Edmunds. Tomato diseases and disorders: Iowa state university. *University Extension*, 2005.

[110] Mohamed Rakha, Ndeye Bouba, Srinivasan Ramasamy, Jean-Luc Regnard, and Peter Hanson. Evaluation of wild tomato accessions (solanum spp.) for resistance to two-spotted spider mite (tetranychus urticae koch) based on trichome type and acylsugar content. *Genetic Resources and Crop Evolution*, 64:1011–1022, 2017.

[111] R Srinivasan. Safer tomato production techniques. *A field guide for soil fertility and pest management. AVRDC Publication. Taiwan*, 2010.

[112] Joicy Aparecida Alves Chaves, Lillian Matias Oliveira, Leandro Castro Silva, Bruno Nascimento Silva, Carla Silva Dias, Jonas Alberto Rios, and Fabrício Ávila Rodrigues. Physiological and biochemical responses of tomato plants to white mold affected by manganese phosphite. *Journal of Phytopathology*, 169(3):149–167, 2021.

[113] Ravi Shankar, Seema Harsha, and Raj Bhandary. A practical guide to identification and control of tomato diseases. *Tropica Seeds PVT Ltd*, 2014.

[114] Loyani K Loyani, Karen Bradshaw, and Dina Machuve. Segmentation of tuta absoluta's damage on tomato plants: A computer vision approach. *Applied Artificial Intelligence*, 35(14):1107–1127, 2021.

[115] Arjen ten Have, Wietse Mulder, Jaap Visser, and Jan AL van Kan. The endopolygalacturonase gene bcpg1 is required for full virulence of botrytis cinerea. *Molecular Plant-Microbe Interactions*, 11(10):1009–1016, 1998.

[116] Albert Khakimov, I Salakhutdinov, A Omolikov, and Samad Utaganov. Traditional and current-prospective methods of agricultural plant diseases detection: A review. In *IOP Conference series: earth and environmental science*, volume 951, page 012002. IOP Publishing, 2022.

[117] Bulent Tugrul, Elhoucine Elfatimi, and Recep Eryigit. Convolutional neural networks in detection of plant leaf diseases: A review. *Agriculture*, 12(8):1192, 2022.

[118] Jun Liu and Xuewei Wang. Plant diseases and pests detection based on deep learning: a review. *Plant Methods*, 17:1–18, 2021.

[119] S Mohana Saranya, RR Rajalaxmi, R Prabavathi, T Suganya, S Mohanapriya, and T Tamilselvi. Deep learning techniques in tomato plant–a review. In *Journal of Physics: Conference Series*, volume 1767, page 012010. IOP Publishing, 2021.

[120] Javaid Ahmad Wani, Sparsh Sharma, Malik Muzamil, Suhaib Ahmed, Surbhi Sharma, and Saurabh Singh. Machine learning and deep learning based computational techniques in automatic agricultural diseases detection: Methodologies, applications, and challenges. *Archives of Computational Methods in Engineering*, 29(1):641–677, 2022.

[121] Mamillapally Nagaraju and Priyanka Chawla. Systematic review of deep learning techniques in plant disease detection. *International journal of system assurance engineering and management*, 11:547–560, 2020.

[122] Mohammed Brahimi, Kamel Boukhalfa, and Abdelouahab Moussaoui. Deep learning for tomato diseases: classification and symptoms visualization. *Applied Artificial Intelligence*, 31(4):299–315, 2017.

[123] Belal AM Ashqar and Samy S Abu-Naser. Image-based tomato leaves diseases detection using deep learning. 2018.

[124] Mohit Agarwal, Abhishek Singh, Siddhartha Arjaria, Amit Sinha, and Suneet Gupta. Toled: Tomato leaf disease detection using convolution neural network. *Procedia Computer Science*, 167:293–301, 2020.

[125] Akshay Kumar and M Vani. Image based tomato leaf disease detection. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–6. IEEE, 2019.

[126] Murk Chohan, Adil Khan, Rozina Chohan, Saif Hassan Katpar, Muhammad Saleem Mahar, et al. Plant disease detection using deep learning. *International Journal of Recent Technology and Engineering*, 9(1):909–914, 2020.

[127] Yang Zhang, Chenglong Song, and Dongwen Zhang. Deep learning-based object detection improvement for tomato disease. *IEEE access*, 8:56607–56614, 2020.

[128] Baidu pictures - discover the colorful world. https://image.baidu.com/.

[129] Ipm images: The source for agriculture and pest management pictures. https://www.ipmimages.org/index.cfm.

[130] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[131] Eustace M Dogo, Oluwatobi J Afolabi, and Bhekisipho Twala. On the relative impact of optimizers on convolutional neural networks with varying depth and width for image classification. *Applied Sciences*, 12(23):11976, 2022.

[132] Christophe Pere. What are loss functions?. after the post on activation functions... — by christophe pere — towards data science. https://towardsdatascience.com/what-is-loss-function-1e2605aeb904.

[133] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*, 2020.

[134] Prabira Kumar Sethy, Nalini Kanta Barpanda, Amiya Kumar Rath, and Santi Kumari Behera. Deep feature based rice leaf disease identification using support vector machine. *Computers and Electronics in Agriculture*, 175:105527, 2020.

[135] Md Delwar Hossain, Hideya Ochiai, Doudou Fall, and Youki Kadobayashi. Lstm-based network attack detection: performance comparison by hyper-parameter values tuning. In *2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, pages 62–69. IEEE, 2020.

[136] Sebastian Raschka. An overview of general performance metrics of binary classifier systems. *arXiv preprint arXiv:1410.5330*, 2014.

[137] AS Saabith, MMM Fareez, and T Vinothraj. Python current trend applications-an overview. *International Journal of Advance Engineering and Research Development*, 6(10), 2019.

[138] anaconda about-us. https://www.anaconda.com/about-us.

[139] Bernadette M Randles, Irene V Pasquetto, Milena S Golshan, and Christine L Borgman. Using the jupyter notebook as a tool for open science: An empirical study. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 1–2. IEEE, 2017.

[140] Android studio. https://developer.android.com/studio.

[141] fire. https://firebase.google.com/.

[142] What is tensorflow? definition from techtarget. https://www.techtarget.com/searchdatamanagement/definition/TensorFlow, 2017.

[143] Keras Team. Keras: Deep learning for humans. https://keras.io/.

[144] https://numpy.org/about/.

[145] About - opencv. https://opencv.org/about/.

[146] Matplotlib documentation – matplotlib 3.7.1 documentation. https://matplotlib.org/stable/index.html.

[147] János Simon, Imre Petkovic, Djerdji Petkovic, and Ármin Petkovics. Navigation and applicability of hexa rotor drones in greenhouse environment. *Tehnički vjesnik*, 25(Supplement 2):249–255, 2018.

[148] Sander Krul, Christos Pantos, Mihai Frangulea, and João Valente. Visual slam for indoor livestock and farming using a small drone with a monocular camera: A feasibility study. *Drones*, 5(2):41, 2021.

[149] CS Arvind, K Prajwal, Amrut C Patil, A Ashwin Kumar, A Sreedevi, and R Jyothi. Low-altitude unmanned aerial vehicle for real-time greenhouse plant disease monitoring using convolutional neural network. In *Soft Computing for Problem Solving: Proceedings of SocProS 2020, Volume 2*, pages 63–76. Springer, 2021.