الجمهورية الجزائرية الديمقراطية الشعبية

People's Democratic Republic of Algeria

وزارة التعليم العالي والبحث العلمي

Ministry of Higher Education and Scientific Research

جامعة الإخوة منتوري قسنطينة 1

Frères Mentouri Constantine 1 University

Université Frères Mentouri Constantine 1

University Frères Mentouri Constantine جامعة الاخوة منتوري

Faculty of natural and Life Sciences كلية علوم الطبيعة و الحياة

Department of applied biologyقسم البيولوجيا التطبيقية

Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of Master in Bioinformatics

---

AGRILINK : A platform for plant pathogens detection and treatment

---

Submitted by:
Djoumana Khelifi.
Inès Bentounsi

Sustained on : 10/07/2023

Board of Examiners:

Chairman: Dr. M.S DAAS

Supervisor: Dr. H. CHEHILI

Examiner: Mr. K. KELLOU

Incubator: Pr. I. BELLIL

CATI: Dr. S. BETINA

Socio-economic partner: Mr. R. Bouarroudj, Agrichem Company

Academic year : 2022/2023

ACKNOWLEDGEMENTS

On writing this final dissertation, we would like to express our gratitude, first and foremost, to the Almighty God for granting us the opportunity to study and complete this work.

We would like to extend our gratitude and sincere thanks to Dr. CHEHILI Hamza for agreeing to supervise us on this subject. We also thank him for his guidance, valuable advice, and availability throughout the period of writing this dissertation.

Our heartfelt thanks go to: Dr. Daas Med Skander and Mr. Kamel Kellou for accepting to chair this committee and the honor of examining this modest work.

To all those who have contributed directly or indirectly to the development of this dissertation, Dr. Bouarroudj Wissem and Dr. Belbekri Adel ,we extend our sincere appreciation

A special thanks to Dr .Benouar Amine for the great work and for his efforts, we are so grateful for his patient , consistency and all the support we got from Qatar to algeria through his help .

We would also like to express our deep gratitude to all the professors in the Department and everyone who has contributed to our education and training.

A heartfelt thank you to our parents for their love, guidance, and unwavering support, both morally and financially, which enabled us to pursue the studies we desired and, consequently, complete this dissertation. We are deeply grateful for their constant encouragement and sacrifices, which have been instrumental in our academic achievements. Their belief in our abilities and their unwavering presence in our lives have been a source of strength and motivation throughout this journey. We acknowledge with profound

appreciation the significant role they have played in shaping our academic and personal development.

Résumé :

L'objectif de ce travail est de relever les défis liés à la gestion des cultures et à la détection des phytopathogènes en développant une plateforme qui permet aux agriculteurs d'obtenir des diagnostics rapides et précis grâce à l'utilisation de l'intelligence artificielle. En facilitant la détection précoce et le diagnostic précis des phytopathogènes, notre plateforme vise à réduire la dépendance aux produits chimiques en agriculture, contribuant ainsi à des pratiques agricoles durables.

Elle fonctionne également comme un centre de connexion entre les agriculteurs, les fournisseurs de produits phytosanitaires et les experts agricoles, favorisant la collaboration et le partage des connaissances au sein de la communauté des acteurs du secteur agricole. Grâce à cette plateforme, les agriculteurs peuvent accéder à des conseils d'experts et à des recommandations, partager les meilleures pratiques et prendre des décisions éclairées concernant la protection des cultures et les stratégies de gestion.

Abstract:

The objective of this work is to address the challenges in crops management and phytopathogen detection by developing a platform that enables farmers to obtain quick and accurate diagnoses using artificial intelligence. By facilitating early detection and precise diagnosis of phytopathogens, our platform aims to reduce the reliance on chemical products in agriculture, contributing to sustainable farming practices.

This platform also functions as a center that connects farmers with phytosanitary product providers and agricultural experts providing collaboration and knowledge sharing that will foster a sense of community among stakeholders in the agricultural sector. Through this platform, farmers can access expert advice and recommendations, share best practices, and make informed decisions about crop protection and management strategies.

[1]

[1] Precision agriculture , pathogenes detection , artificial intelligence diagnostic

ملخص

الهدف من هذا العمل هو التعامل مع التحديات المتعلقة بإدارة المحاصيل واكتشاف الأمراض النباتية من خلال تطوير منصة تمكّن المزارعين من الحصول على تشخيصات سريعة ودقيقة باستخدام الذكاء الاصطناعي. من خلال تيسير الكشف المبكر والتشخيص الدقيق للأمراض النباتية، تهدف منصتنا إلى الحد من الاعتماد على المنتجات الكيميائية في الزراعة، مما يساهم في الممارسات الزراعية المستدامة.

تعمل المنصة أيضًا كمركز يربط المزارعين بمزودي المنتجات الزراعية والخبراء الزراعيين، مما يوفر التعاون وتبادل المعرفة ويعزز التواصل بين أصحاب المصلحة في قطاع الزراعة. من خلال هذه المنصة، يمكن للمزارعين الوصول إلى نصائح الخبراء، التوصيات واتخاد قرارات صائبة بشأن حماية وإدارة المحاصيل.

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

PA: Precision Agriculture.

AI: Artificial intelligence.

ML: Machine learning.

DL: Deep learning.

ANN: Artificial neural network.

CNN: Convolutional neural network.

MVT: Model-View-Template.

Contents

INTRODUCTION

Introduction

Water, food, and plants are regarded as the most vital aspects of livability; they serve as the basis for our existence and ensure both animal and human life and well-being. Among these, agriculture plays a crucial role in producing the majority of the world's food supply.

However, population increase and resource depletion have combined to pose a threat to humankind. Over time, food security gradually deteriorates because of this persistent danger that has been further exacerbated by phytopathogens and the conventional practices employed to combat them in the agricultural sector. This may result in a lack of food security and large crop losses, which would cut down on the total supply of food supplies.

Therefore, the agricultural sector has embraced smart farming practices including precision agriculture (PA), utilizing various technologies to mitigate these risks and enhance overall productivity and sustainability and ensure food security.

In our study, we will focus on the detection of phytopathogens and plant diagnostics using artificial intelligence (AI) through a platform built on the Django framework. This approach aligns with the principles of precision agriculture, which aims to reduce agrochemical inputs and mitigate the negative environmental effects of traditional farming practices. By integrating AI technology into our platform, we aim to enhance the efficiency and accuracy of plant disease identification, thereby contributing to the overall goal of sustainable and environmentally friendly agriculture [3].

The primary objective of this study is to optimize the management of agricultural parcels by implementing advanced techniques for the detection of phytopathogens while minimizing the reliance on chemical interventions. Furthermore, it aims to consolidate these functionalities within a platform, facilitating the communication and collaboration among farmers, suppliers of plant protection products, and domain experts.

The planning of our thesis is as follows: it is composed of four chapters; the first chapter represents a state of the art about precision agriculture. This chapter provides an overview of precision agriculture, discussing its significance and impact on the agricultural sector worldwide. It explores the transition from traditional farming practices to intelligent farming, highlighting the benefits and advancements associated with precision agriculture.

In the second chapter, the focus is on the role of artificial intelligence (AI) in intelligent agriculture. It reviews relevant literature and research works that highlight the application of AI techniques in precision agriculture while in the third chapter the focus is in the model architecture used in our work explaining the design and components of the model in detail, providing insights into its structure and functioning.

Finally, the fourth chapter focuses on the development and conception of our Django-based platform Agrilink for plant diagnosis. It discusses the design, features, and technologies utilized in the platform.

Chapter 1 :

Precision Agriculture.

Precision agriculture

1. Definition of agriculture

Agriculture is a comprehensive term that encompasses the various ways in which crop plants and domestic animals sustain the global human population by providing essential food and other products. It encompasses a diverse range of activities integral to agriculture, such as cultivation, domestication and other different forms of livestock management. While agriculture is sometimes narrowly defined as crop cultivation excluding animal husbandry, it generally implies both activities. In its broadest sense, agriculture is the science and art of cultivating the soil, including the related practices of harvesting crops and raising livestock [1].

1.2 Cultivation

Cultivation refers to the practice of humans actively managing the growth and development of specific plants. It involves preparing the soil, sowing seeds, and manipulating the plant environment to ensure optimal conditions for growth. Cultivation techniques vary, from basic methods like clearing vegetation and tilling the soil to more advanced approaches such as adding nutrients and implementing crop rotations. By actively engaging in cultivation, humans exert control over the plant's life cycle and increase the likelihood of obtaining desired yields [1].

2. Types of agriculture

Various types of agriculture address different farming requirements and circumstances.
Here are some examples:

2.1 Conventional agriculture

Any agricultural system that makes use of chemical inputs is referred to as "conventional agriculture" in general. Conventional agriculture does not disallow any external inputs that may be advantageous for its production [22].

2.2 Extensive agriculture

Extensive agriculture is an agricultural system in which large farms are cultivated with a moderately lower amount of labor and inputs. It is less mechanized and characterized by relatively low yields.

Extensive agriculture is practiced on very large areas divided into sections. It prioritizes natural resources to promote crop growth and requires less specialized labor.

2.3 Intensive agriculture

Intensive agriculture is an agricultural system that aims to maximize production on limited land by using inputs and labor in order to generate maximum profits within short timeframes.

This mode of operation requires hard work, efficient mechanical tools, and the use of various fertilizers. Intensive farming is less commonly used in Africa due to a lack of financial resources and is limited to a few high-density areas [23].

2.4 Subsistence agriculture

Subsistence farming is a style of farming in which the primary focus is on growing crops or raising livestock for the farmer and their family's sustenance. Typically, there is little to no excess for selling or trade. Preindustrial agricultural communities across the world have long used this type of farming. It involves minimal use of modern techniques and relies on small land holdings and manual labor. The produced output is primarily consumed locally, Crop choices are based on future family needs and. Subsistence farming was prevalent before industrialization, providing sustenance for many people [33].

2.5 Sustainable agriculture

Sustainable agriculture is a multidisciplinary field that tackles pressing challenges including climate change, food and fuel price increases, hunger in impoverished nations, and obesity in affluent nations, water pollution, soil erosion, fertility decline, pest management, and biodiversity depletion. Its primary goal is to produce food and energy in a manner that can be sustained by present and future generations. Sustainable agriculture proposes innovative, eco-friendly approaches that draw upon integrated knowledge from diverse scientific disciplines such as agronomy, soil science, molecular biology, chemistry, toxicology, ecology, economy, and social sciences [34].

2.6 Organic agriculture

Organic agriculture is a farming system that focuses on the preservation and recycling of resources, with the aim of creating more sustainable production systems. It is based on internationally accepted principles and legally binding standards in some jurisdictions. Organic production methods aim to reduce dependence on fossil-fuel inputs while promoting environmentally friendly practices.

In organic agriculture, farmers prioritize natural approaches to fertilize the soil, control pests and diseases, and promote biodiversity. They avoid the use of synthetic chemical pesticides, inorganic fertilizers, and other potentially harmful substances to the environment and human health [35].

2.7 Commercial agriculture

Commercial agriculture refers to the type of farming where food or agricultural products are produced primarily for sale rather than solely for personal consumption. It involves the cultivation of crops and/or raising livestock with the intention of generating profit. Commercial agriculture encompasses various forms of farming, such as dairy, grain, plantation, livestock, fruit, and mixed crop farms. It includes selling produce through different channels like restaurants, supermarkets, wholesale distributors, and local produce stands. The focus of commercial agriculture is to maximize yields while minimizing costs to achieve a higher profit margin [36].

3. Revolutionizing Farming through Intelligent Technologies and Data-Driven Solutions

Agriculture produces the majority of the world's food; it is considered as the backbone of economic systems for developing countries. For decades, agriculture has been related with the production of vital food crops. Moreover, healthy food systems are crucial aspects in achieving global development, which is why agricultural development is one of the primary levers for ending poverty. However, it has been difficult to provide and ensure food security to the amount of population that the world has arrived to with traditional farming, that is why agriculture took a step towards and adopted intelligent ways.

A first green revolution began, consisting of the use of fertilizers and new machinery that increased yields, and then a second revolution was introduced, which was the introduction of new crop varieties, more fertilizers, pesticides and other technologies that required less human effort. Although crop productivity has increased due to this revolution but yields of the major crops are expected to decline due to rising temperatures , changes in water availability and pest diseases, a third green revolution was then needed : INTELLIGENT AGRICULTURE .

Intelligent agriculture harnesses the power of information and communication technologies to drive productivity, efficiency, and profitability in agricultural operations. However, simply implementing these technologies is not enough; they must be coupled with robust data mining tools to extract relevant, reliable, and actionable information. By collecting and analyzing various data types, including meteorological, agronomic, bio-aggressor risks, and economic factors, precision agriculture emerges as a key solution [38].

The agricultural sector faces numerous challenges, which significantly affect crop productivity, with crop diseases being a prominent threat. Throughout history, innovative solutions devised by humans have successfully combated such challenges. Today, maintaining the health and well-being of crops is paramount. Precision agriculture plays a crucial role in achieving highly satisfactory and healthy outcomes, especially in tasks like early detection of plant pathogens or accurately estimating the optimal quantity of fertilizers required for specific plots. By minimizing waste and avoiding damage to the fields, precision agriculture provides effective solutions to these problems.

In summary, intelligent agriculture integrates cutting-edge technologies and data mining tools to drive productivity and efficiency. By addressing challenges like crop diseases and optimizing resource usage, precision agriculture emerges as a vital approach to ensure healthier crops and sustainable farming practices.

4. Precision agriculture (PA)

Precision agriculture is a transformative approach that offers solutions to reduce agrochemical inputs and mitigate the negative environmental effects of traditional farming practices. By implementing precision agriculture techniques, farmers can achieve several key benefits. Firstly, they can experience economic advantages through reduced costs by precisely applying agricultural inputs based on field conditions. Secondly, precision agriculture enables targeted management of field variability, leading to increased yields. Lastly, it promotes a favorable environmental impact by ensuring precise and controlled application of agrochemical products.

Precision agriculture comprises a set of technologies that combines sensors, information systems, enhanced machinery, and informed management to optimize production, it provides a means to monitor the food production chain and manage both the quantity and quality of agricultural produce[3].

4.1 Economic effects of precision agriculture

Precision agriculture (PA) techniques offer a wide range of benefits, both in terms of quantity and quality. These benefits, as defined by experts, include:

- Reduced driver fatigue: Guidance systems assist in maintaining precise paths, reducing the efforts required from drivers and minimizing fatigue.
- Increased yields: Precision agriculture enables targeted management of field variability, leading to improved yields.
- Reduced costs of work operations: By optimizing resource usage, precision agriculture leads to cost savings in labor, equipment, chemical inputs (such as plant protection products, fertilizers, and seed stock), and fuels.
- Increased productivity: The ability to operate at higher speeds enhances overall productivity.
- Better quality: With the assistance of precision agriculture technologies, drivers can focus their attention elsewhere, resulting in improved product quality.
- Lower adverse environmental impact: Precision agriculture enables precise application of inputs, reducing environmental impacts associated with excessive agrochemical usage.
- Ability to work at night and under reduced visibility: Precision agriculture technologies, such as advanced sensors and guidance systems, allow for efficient operations during nighttime or under conditions of limited visibility.
- Time savings: The streamlined and optimized processes of precision agriculture contribute to time savings throughout various farming operations [2].

These benefits highlight the comprehensive advantages offered by precision agriculture, encompassing improved productivity, cost reduction, environmental sustainability, and enhanced operational efficiency.

4.2 Enhancing Food Security through Precision Agriculture

Food security, as defined by the United Nations in 1996, encompasses the physical and economic access to sufficient, safe, and nutritious food that meets individual's dietary needs and preferences for an active and healthy life (FAO, 2012). Beyond its economic impacts, precision agriculture plays a pivotal role in addressing food security challenges.

Increasing the usage of crops with high nutritional content is one of the most difficult tasks in achieving food security. Even well nourished people suffer from diseases caused by nutritional deficiencies, it is crucial to grow high-quality crops that fulfill good food quality requirements in order to counteract the development of illnesses caused by nutritional deficits. A plentiful supply of healthy food not only reduces hunger but also helps to avoid various malnutrition-related disorders. In fact, Climate, crop type and cultivar, cultural methods, management approaches, and fertilizer use all have an impact on crop nutritional quality [4].

Consequently, precision agriculture serves three key purposes. Firstly, it seeks to maximize the use of existing resources, resulting in enhanced profitability and sustainability in agricultural operations. Secondly, it works to reduce negative environmental consequences by precise and focused resource application. Lastly, it aims to improve the whole work environment and social elements of farming, ranching, and associated professions. [5].

By incorporating precision agriculture techniques, the agricultural sector can play a vital role in achieving global food security goals. The ability to efficiently utilize resources, minimize waste, and improve crop quality and productivity contributes to ensuring that all individuals have reliable access to safe, nutritious, and sufficient food for their well-being.

4.3. The difference between intelligent agriculture and precision agriculture

Precision agriculture is largely concerned with optimizing agricultural operations via the use of technology, data analysis, and precise resource management in order to increase output, decrease costs, and reduce environmental effect.

Intelligent agriculture, on the other hand, is a wider notion that incorporates precision agriculture as one of its components. It extends beyond resource management accuracy to include intelligent decision-making processes, automation, and real-time monitoring for more efficient and sustainable agricultural systems.

To summarize, precision agriculture is a subfield of smart agriculture that focuses on precise resource management and data-driven decision making to maximize productivity while minimizing environmental impact. Smart agriculture, on the other hand, goes beyond a precise approach to building a more integrated, data-intensive and autonomous agricultural framework by integrating modern technologies and intelligent systems.

Chapter 2 :

AI in precision agriculture

AI in Precision Agriculture

1. Artificial intelligence (AI)

The study and construction of algorithms that can mimic human activities is known as artificial intelligence. An intelligent system can take many different shapes. It is capable of becoming indistinguishable from a person, which is known as general artificial intelligence. It might also be a voice assistant like Siri, Alexa, Cortana, or Google Assistant. It might be an automated vehicle, similar to Tesla vehicles. AI also includes online store suggestions and non-playable characters (NPCs) in video games. Intelligent systems are capable of making judgments to solve issues without the need for human interaction, which is where their intelligence originates from. [15].

1.1. Machine learning (ML)

ML emerges at the crossroads of computer science and statistics, it consists of employing algorithmic methodologies to accomplish task-specific objectives without the need for explicit programming instructions. Its primary function involves the discernment of data patterns and subsequent generation of predictions based on these discerned patterns. Importantly, ML encompasses two distinct processes: supervised learning and unsupervised learning, each characterized by notable disparities. Supervised learning entails the utilization of labeled training data to train models, enabling them to accurately predict or classify new instances. Conversely, unsupervised learning operates on unlabeled data, aiming to uncover inherent patterns and structures within the dataset without relying on predefined categories or labes [16].

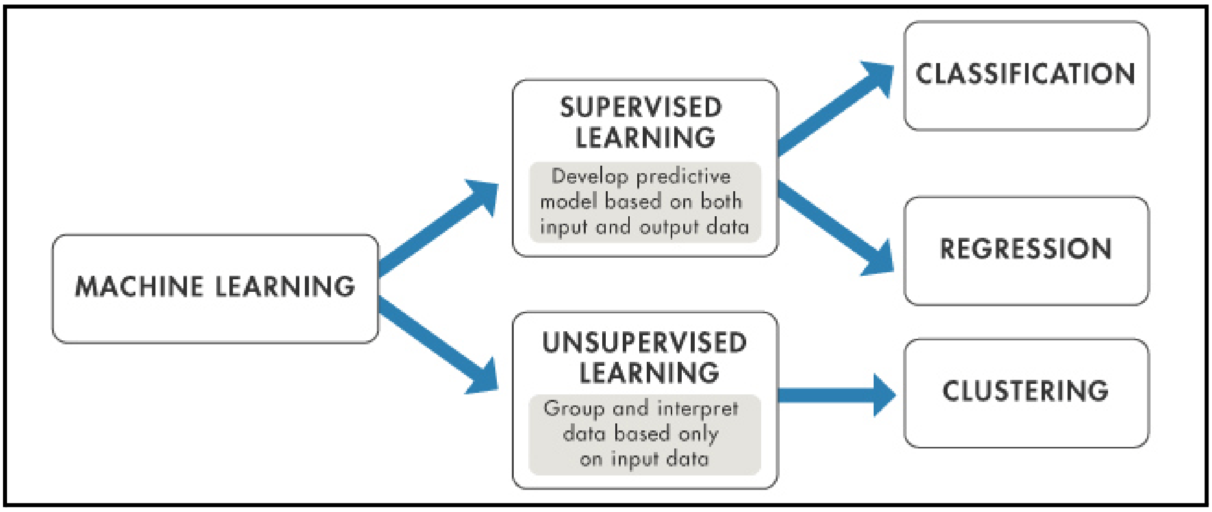

Figure 1. The difference between supervised and unsupervised learning [26].

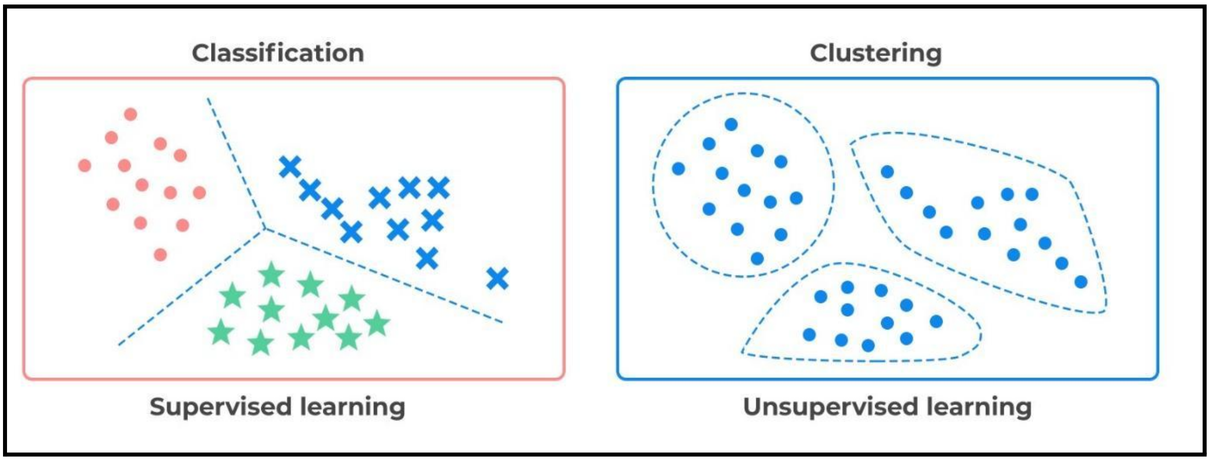

Figure 2. Supervised learning vs unsupervised learning [27].

1.2. Deep learning (DL)

DL is a computational approach that uses advanced algorithms to analyze and extract insights from data. Within its applications, deep learning utilizes artificial neural networks (ANNs) to address intricate problem domains.

1.3. Artificial neural network (ANN)

ANNs are designed to simulate the behavior of human brain neurons and are particularly adept at handling complex tasks. Notably, deep learning thrives on large-scale datasets as it benefits from the ability to learn complicated patterns and representations. As such, it can be regarded as a specialized subset of machine learning, with its distinctive focus on hierarchical representations and intricate feature learning [13].

The relationships among artificial intelligence (AI), machine learning (ML), and deep learning (DL) can be illustrated by the following figure, highlighting the subtle distinctions between these concepts :

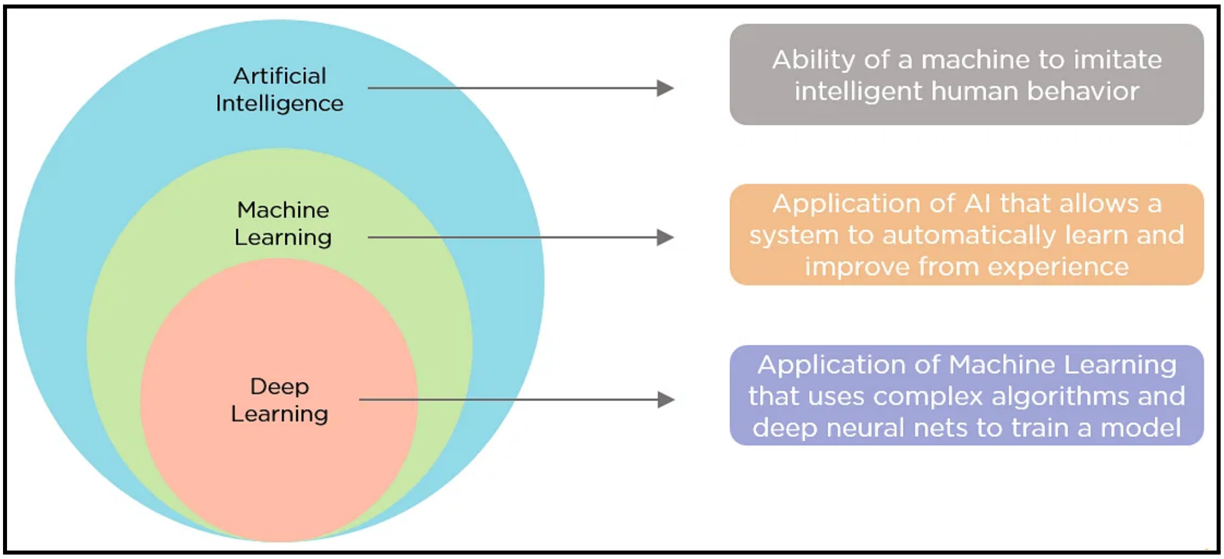

Figure 3. Representation of the difference between artificial intelligence, machine learning and deep learning. [14]

1.5. Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a deep learning model specifically designed for processing data with a grid-like structure, such as images. It takes inspiration from the organization of the visual cortex in animals and aims to automatically learn spatial hierarchies of features, progressively detecting low-level to high-level patterns.

The CNN architecture consists of three primary types of layers: convolutional, pooling, and fully connected layers. The convolutional and pooling layers are responsible for extracting relevant features from the input data. Through the process of convolution, the network applies filters to capture different patterns within the data. The pooling layers downsample the feature maps, reducing their dimensionality while preserving important information.

The final component of a CNN is the fully connected layer, which takes the extracted features and maps them to the desired output, such as classification labels. This layer combines the learned features and applies further transformations to make predictions or decisions based on the input data.

By utilizing the hierarchical arrangement of convolutional, pooling, and fully connected layers, CNNs can effectively learn complex patterns and spatial relationships within the input data, making them particularly suited for tasks such as image classification, object detection, and image segmentation.[20]

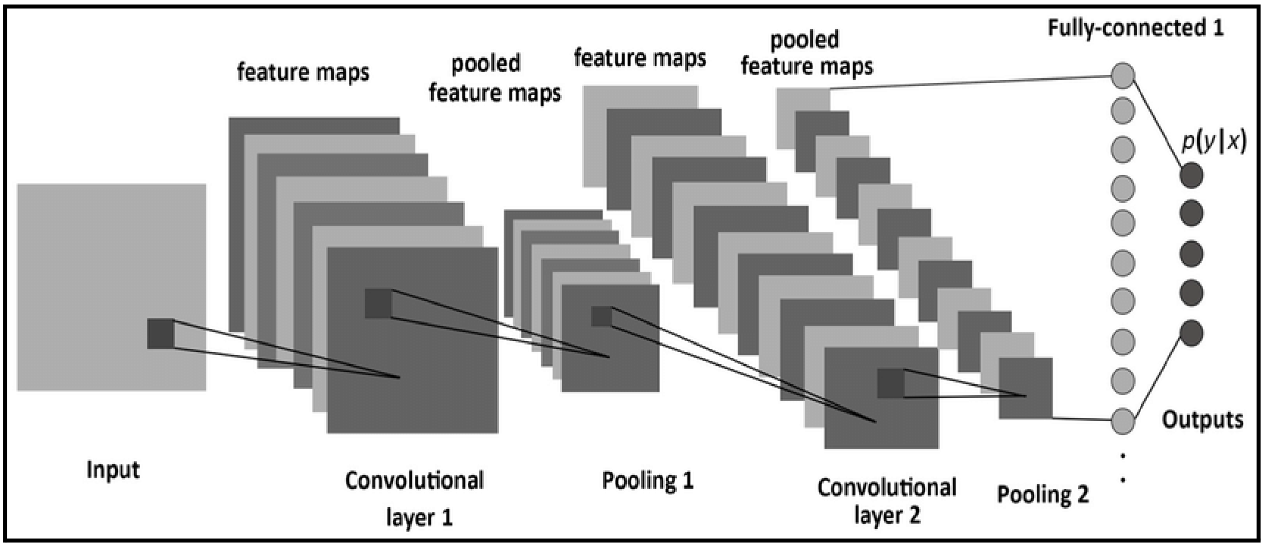

Figure 4. Classic architecture of a convolutional neural network.[28]

2. From traditional to smart agriculture

In the present era, many countries persist in adhering to traditional farming practices characterized by the use of conventional equipment. However, this approach has built-in limits when it comes to maintaining food security and advancing the world economy.

On the other hand countries considered as developed ones are adopting smart agricultural as the first and main way to control and manage fields, using smart equipment , machine learning and deep learning mobile and web platforms .Moreover , Agriculture encounters various constraints, including climate change, modifications in ecosystems, and crop diseases. By its very nature, agriculture is reliant on living organisms—both plants and animals—that undergo evolution and are intricately interconnected with their environment, encompassing factors such as climate, soil composition, and pathogens.

Agriculture is built and reasoned on multiple scales: the parcel, the farm, the country and the world... Today, the farmer and all of his prescribers cannot reason alone in their decision-making. They need to know the local and international trends that will influence markets, food prices, relationships with consumers [12].Traditionally, visual examination by experts has been carried out to diagnose plant diseases and biological examination is the second option, if necessary. However, there is a risk for error due to subjective perception, in this context; various spectroscopic and imaging techniques have been studied for detecting plant diseases.

In recent years, the advancement of computer technology has facilitated the widespread adoption of machine learning techniques for training and detecting plant diseases. As a viable alternative, machine learning offers a satisfactory approach for the accurate detection of plant diseases [9].

In addition, Deep Learning offers solutions to better anticipate these trends and help farmers make their decisions because it can integrate massive data from heterogeneous sources [12].

3. Deep Learning Applications in Agriculture :

As a proposed solution, numerous developers have embraced the concept of smart farming by developing web and mobile applications. Several of these applications have garnered recognition and are featured in various articles, as follows:

3.1. The cassava model of PlantVillage Nuru :

The cassava model of PlantVillage is a deep learning model to identify foliar symptoms of diseases in cassava. A CNN model was trained using an imagery dataset of 720 diseased leaflets in an agricultural field in Tanzania. The developed model could detect seven classes of healthy and infected cassava leaflets, namely, healthy, brown streak disease, mosaic disease, green mite damage, red mite damage, brown leaf spot, and nutrient deficiency [7].

The user is directed to point the phone's camera onto a leaf that does not look healthy and ensure that the image is in focus prior to analysis. Once the image is in focus, the user can start the analysis and the app will display boxes indicating the condition it has identified on individual leaflets. Once the user has finished inspecting the whole plant the app provides the user with an overall diagnosis of the condition of the plant followed by advice on management of the disease or pest it has detected .

They worked with a Convolutional Neural Network (CNN) model that can feed images directly from farmer's mobile devices. The model then performs object detection and semantic segmentation, and displays the disease category along with the confidence percentage and classification time have taken to process the image. In addition, the application displays the confidence percentage and classification time taken to process the image.

3.2. Ricetalk, Rice blast detection using internet of things and artificial intelligence technologies:

Using a combination of the Internet of Things (IoT) and Artificial Intelligence (AI) technologies to detect the rice blast disease in its early stages. An IoT platform, called RiceTalk, was developed to detect rice blast utilizing non-image IoT devices, which generate sensing data from soil cultivation. The sensed data could be automatically trained and analyzed by a CNN model in real time. RiceTalk achieved an average prediction accuracy of 89.4% for detecting a rice blast disease in the natural agricultural field [8].

3.3. Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks:

An apple leaf disease detection approach based on the Mask Region-based CNN. A dataset containing 2029 images of diseased apple leaves is used to train a CNN model for detecting the common apple diseases. The developed model could detect five disease classes: Alternaria leaf spot, brown spot, mosaic, gray spot, and rust. Given the relatively small dataset used for training the CNN model, the classification accuracy was calculated to be 78.8%; the proposed model also exhibits strong detection performance and robustness [9].

3.4. Plantix:

Plantix (Berlin, Germany), is a diagnosis application that detects diseases, pests, and nutritional deficiencies using an image of the plant to be analyzed.

The application uses image recognition and deep learning (DL) algorithms to detect about 400 damage occurrences in 30 crop types. It also offers a list, divided by regions, with the most common diseases. The image is sent to the cloud where it is processed and a result is generated. Besides receiving the individual diagnosis, the user can also view information about conventional treatments, alternative treatment options and preventive measures [10].

In this work they used the PlantVillage dataset which has 38 classes of diseases of various plants along with healthy leaves images as well.

TensorFlow and Keras are the used Python libraries for preprocessing and processing of images by applying different models in Plantix. Tensor flow is used for machine learning and it is open source [19].

3.5. E-agree:

An image processing system to detect plant diseases based on their leaves, a marketplace, which helps farmers sell/buy products online, and a market rate guide that helps farmers to gather information about market rates. It also provides a weather reporting system that plays a crucial role in decision-making, as well as soil types information to guide farmers in deciding which crop type is the most appropriate to their fields [11].

In addition, there are also B2B (business-to-business) platforms available:

3.6. Linkinfarm:

Linkinfarm, a B2B startup, was established based on the realization that agricultural equipment expenses pose a significant burden for farms, with projections indicating further cost increases in the future. Despite the existence of alternatives to equipment ownership, such options remain largely untapped by farmers. Since 2017, Linkinfarm has been dedicated to developing services and solutions tailored to the agricultural sector, primarily targeting farmers. The aim is to simplify and enhance the reliability of delegating agricultural tasks, thereby supporting farmers in maintaining a productive and profitable family farm model. By advocating for a systematic adoption of partial or complete delegation in land management, Linkinfarm strives to enable farmers to optimize their operations effectively.[18]

3.7. Farm21:

As a B2B service, Linkinfarm offers a comprehensive platform that grants users swift access to essential information for optimizing their farming operations. The Farm21 dashboard serves as a centralized hub, consolidating data from various sources including scouts, weather reports, satellites, crop analytics, and sensors. This integration of diverse data sets allows for informed, data-driven decision-making in daily operations. With a holistic, 360-degree view of fields and crops, farmers can effectively prioritize crucial tasks such as irrigation, disease control, and labor management. By providing this extensive suite of features, Linkinfarm empowers businesses in the agricultural industry to streamline their operations and maximize efficiency. [17].

4. Summary: integrating detection and provision in our platform

In summary, the application of deep learning and image processing methods for plant disease identification has the potential to completely transform the agricultural industry. This advancement enables the timely identification and treatment of plant diseases, leading to increased crop yields and reduced food waste.

Furthermore , within the landscape of agricultural platforms, some are made for the purpose of identifying plant leaf diseases using image processing methods, while others act as commercial platforms that market pesticides and efficient plant treatments.Inspired by the existing offerings, our vision is to create a comprehensive platform that combines both detection and provision services. This platform will unite professionals and reputable suppliers within the industry. In terms of accuracy, our objective is to establish a new standard by aiming to outperform current systems in terms of precision.

Chapter 3:

A smart model for plant disease identification

A smart model for plant disease identification
Model architecture

I. MATERIAL AND METHODS OF THE MODEL

1. Material

1.1. Used data:

The PlantVillage dataset, being the largest and extensively researched plant disease dataset, comprises over 54,000 images of leaves. Notably, when applying the convolutional neural network trained on the PlantVillage dataset to various online datasets, a significant decline in accuracy was observed.

The dataset consists of 54,305 individual leaf images, each measuring 256 pixels by 256 pixels and captured in RGB format. These images cover 14 different crop species. The dataset is divided into 38 classes, which include species with diseases and species that are healthy. The leaves were detached from the plants and positioned against a background of either grey or black. All photographs were taken outdoors using a single digital camera, under varying weather conditions ranging from sunny to cloudy days. [25]

Table 1. Class names of the PlantVillage dataset.

Apple apple scab	Grape black rot	Strawberry leaf scorch
Apple blackrot	Grape leaf blight	Strawberry healthy
Apple cedar apple rust	Grape healthy	Tomato bacterial spot
Apple cedar apple rust	Orange huanglongbing	Tomato early blight
Apple healthy	Peach bacterial spot	Tomato late blight
Blueberry healthy	Peach healthy	Tomato leaf mold
Cherry powdery mildew	Bell Pepper bacterial spot	Tomato septoria leaf spot
Cherry healthy	Bell Pepper healthy	Tomato spider mites
Corn gray leaf spot	Potato early blight	Tomato target spot
Corn common rust	Potato late blight	Tomato yellow leaf curl virus
Corn northern leaf blight	Potato healthy	Tomato tomato mosaic virus
Corn healthy	Raspberry healthy	Tomato healthy
Mosaic virus Grape	Black rot Soybean	
Grape esca	Squash powdery mildew	

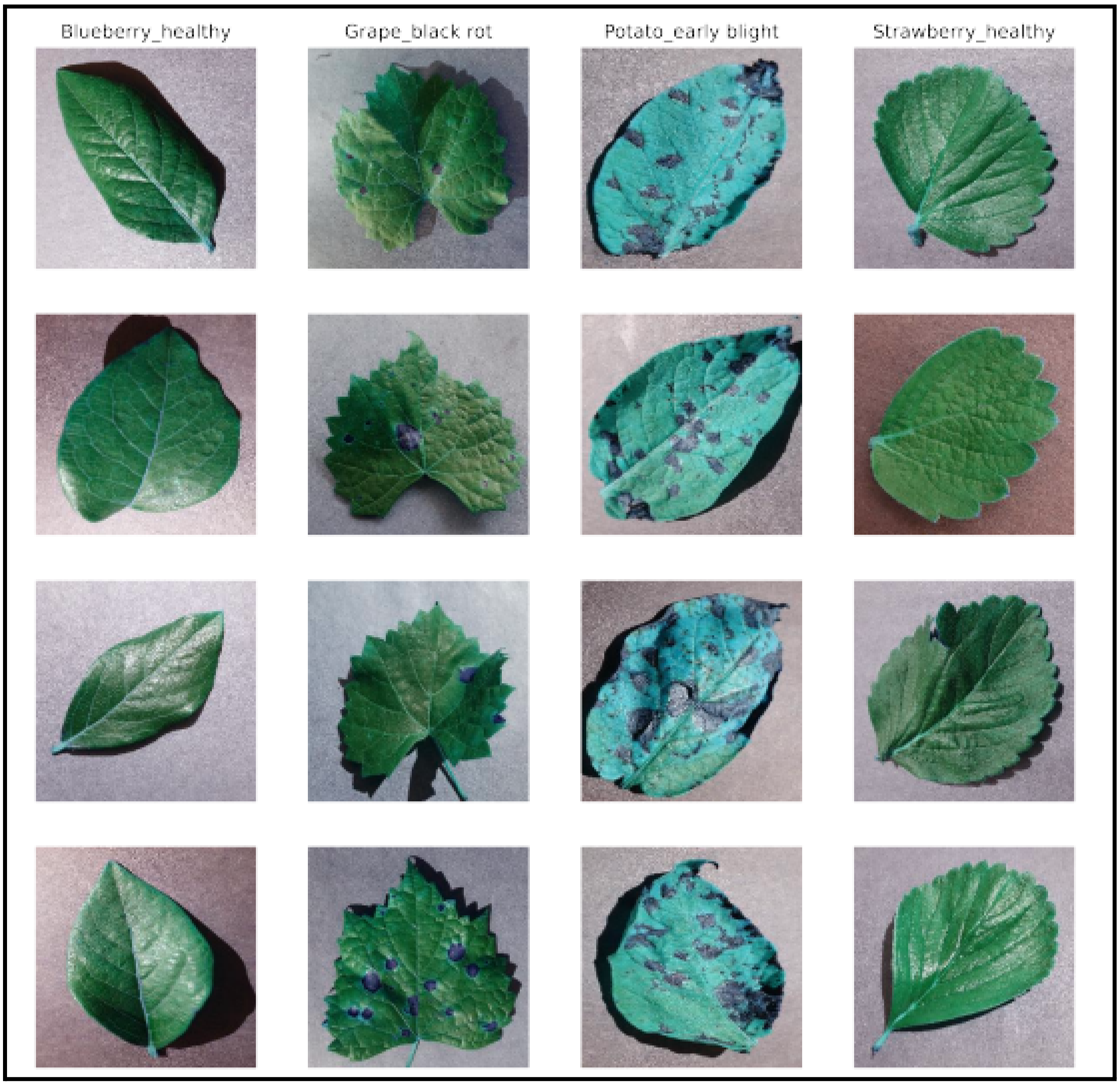

Figure 05. Images from the PlantVillage dataset. Each column belongs to a single class [25].

1.2. Machine configuration: Its specifications are detailed in the table below:

Table 2. Specifications of the used machine.

Computer	Specifications
Processor	CPU 24 Cœurs
Installed RAM	64 Go
Graphic card	Invidia GeForce RTX™ 3090 Ti

1.3. Computer Tools and Libraries: We briefly describe the tools and libraries used to accomplish the work as follows:

- Python[2] 3.9: An open-source, interpreted, high-level programming language designed to be easy to read and write. It is used in the fields of data science and analysis, as well as web development, system automation, and software creation. Python has a large standard library, as well as many third-party libraries for data manipulation, visualization, machine learning, etc.
- Tensorflow[3] : An open-source machine learning library developed by Google. It allows for the construction and training of deep neural networks for a variety of tasks, such as classification, image recognition, speech recognition, language translation, and many more. TensorFlow provides a flexible API for model creation, as well as a graphical interface for visualization of computational graphs. It is also compatible with GPUs for accelerated computations and can be used in conjunction with other Python libraries like NumPy and Pandas.
- Matplotlib[4] : A popular data visualization library in Python. It allows for the creation of a wide variety of charts, plots, and visualizations to visually represent data in 2D and 3D. Matplotlib is a flexible and powerful library widely used in scientific, data analysis, and visualization fields.
- Keras[5] : an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. Keras also gives the highest priority to crafting great documentation and developer guides.

- Numpy[6] : it is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast

[2] https://www.python.org/
[3] https://www.tensorflow.org/
[4] https://matplotlib.org/
[5] https://keras.org/
[6] https://numpy.org/

operations on arrays, including mathematical, logical, shape manipulation and sorting.

2. Method

This section represents the methods used to achieve phytopathogen prediction.

2.1 Data augmentation and preprocessing

Data augmentation and preprocessing are methods for improving the training data for our ResNet-152 model. Here is how it works:

2.1.1. Splitting the validation dataset:

This step involves splitting the data into small batches, including test batches and validation batches, which are processed sequentially. Specifically, in the context of the PlantVillage dataset, there is 80 % validation batches and 20% test batches.

Calculating the number of batches in the validation dataset entails determining the number of batches into which the validation dataset will be divided. The test dataset is then created by taking a subset (1/5) of the validation dataset, and the remainder of the validation dataset is allocated back to the validation dataset by skipping the test dataset.

2.1.2. Data augmentation : this part entails implementing a series of data augmentation transformations to the image, such as RandomFlip ('horizontal') which randomly flips the image horizontally and RandomRotation(0,2) which rotates the image randomly by up to 20%.

2.2. Metrics and plotting functions

Metrics plotting entails viewing the metrics of interest at various phases of the training process; it begins by specifying a set of colors and metrics to be plotted (loss, auc, accuracy, and recall). It also determines the plot figure's size.

2.3. Loading and compiling the model

This step refers to the process of selecting and configuring the architecture of the model that we have chosen to use for our image classification task, which is ResNet-152.

2.3.1 ResNet-152: A deep neural network architecture that introduces a residual learning framework to address the challenges of training deeper networks, it is a variant of the ResNet architecture with 152 layers.

ResNet-152 reformulates the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. It is easy to optimize, and can gain accuracy from considerably increased depth [37].

In this step, the ResNet152 basic model is initialized with pre-trained weights and the input photos are prepared to make sure they are in the right format for the model by applying specific preprocessing operations, such as normalization and scaling.

The parameters of the model are freezed to prevent them from being updated in this phase by setting the attribute 'trainable' to 'false'.

2.3.2 The construction of the model:

Once the data is preprocessed and augmented, the model construction **(Figure 06)** begins. The model is composed of the following layers:

1. The InputLayer: This layer represents the input to the model and expects images of size 300x300 pixels with three colors channels (RGB).

2. The Sequential_5 layer: it acts as an intermediate step in the image-processing pipeline.

3. The tf.operators.getitem_2 (SlicingOpLambda) layer: This layer performs a slicing operation to extract a specific portion of the input data.

4. The tf.nn.bias_add_2 (TFOpLambda) layer: it adds a bias to the input data, similar to the previous model.

5. The resnet152 (Functional) layer: this layer represents the pretrained ResNet152 network. It consists of multiple residual blocks that enable the extraction of complex features from our input images.

6. The GlobalAveragePooling2D layer: this layer performs dimensionality reduction by calculating the average activations across each channel of the input image.

7. The Dropout layer: This layer applies random regularization by randomly deactivating some neurons during training to prevent overfitting.

8. The Dense layer: The last layer of the model is a fully connected layer that generates the classification probabilities for the 38 predefined classes. This layer has 38 output nodes, indicating that the model is designed for multiclass classification.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_8 (InputLayer)        [(None, 300, 300, 3)]     0

 sequential_5 (Sequential)   (None, 300, 300, 3)       0

 tf.__operators__.getitem_2   (None, 300, 300, 3)      0
 (SlicingOpLambda)

 tf.nn.bias_add_2 (TFOpLambd  (None, 300, 300, 3)      0
 a)

 resnet152 (Functional)      (None, 10, 10, 2048)      58370944

 global_average_pooling2d_3   (None, 2048)             0
 (GlobalAveragePooling2D)

 dropout_3 (Dropout)         (None, 2048)              0

 dense_3 (Dense)             (None, 38)                77862

=================================================================
Total params: 58,448,806
Trainable params: 77,862
Non-trainable params: 58,370,944
```

Figure 06. The model summary.

2.4. Training, validating and testing

During this phase, the model is trained using a training dataset for training and a validation dataset for validation, with the initial number of epochs set to 40 ,it is trained for the specified number of epochs on the training dataset. To track development and avoid overfitting, the model's performance is assessed on the validation dataset after each epoch.

The training is started by using the “fit” function of Tensorflow; it optimizes and updates the model’s parameters in a way that minimizes the difference between the predicted output and the actual label.

Then a history object is returned as a result containing information about the training process, such as loss and accuracy metrics for both the training and validation sets at each epoch, it is show in the following figure:

```
637/637 [==============================] - 398s 624ms/step - loss: 0.0816 - tp: 39541.0000 - fp: 965.0000 - tn: 1506008.
0000 - fn: 1188.0000 - accuracy: 0.9733 - precision: 0.9762 - recall: 0.9708 - auc: 0.9991 - val_loss: 0.0597 - val_tp:
10670.0000 - val_fp: 194.0000 - val_tn: 402662.0000 - val_fn: 218.0000 - val_accuracy: 0.9809 - val_precision: 0.9821 -
val_recall: 0.9800 - val_auc: 0.9992
Epoch 38/40
637/637 [==============================] - 400s 628ms/step - loss: 0.0868 - tp: 39495.0000 - fp: 1044.0000 - tn: 150592
9.0000 - fn: 1234.0000 - accuracy: 0.9717 - precision: 0.9742 - recall: 0.9697 - auc: 0.9989 - val_loss: 0.0635 - val_t
p: 10653.0000 - val_fp: 201.0000 - val_tn: 402655.0000 - val_fn: 235.0000 - val_accuracy: 0.9796 - val_precision: 0.9815
- val_recall: 0.9784 - val_auc: 0.9990
Epoch 39/40
637/637 [==============================] - 399s 626ms/step - loss: 0.0854 - tp: 39495.0000 - fp: 1035.0000 - tn: 150593
8.0000 - fn: 1234.0000 - accuracy: 0.9720 - precision: 0.9745 - recall: 0.9697 - auc: 0.9989 - val_loss: 0.0579 - val_t
p: 10682.0000 - val_fp: 182.0000 - val_tn: 402674.0000 - val_fn: 206.0000 - val_accuracy: 0.9822 - val_precision: 0.9832
- val_recall: 0.9811 - val_auc: 0.9993
Epoch 40/40
637/637 [==============================] - 399s 626ms/step - loss: 0.0826 - tp: 39506.0000 - fp: 1023.0000 - tn: 150595
0.0000 - fn: 1223.0000 - accuracy: 0.9719 - precision: 0.9748 - recall: 0.9700 - auc: 0.9990 - val_loss: 0.0621 - val_t
p: 10673.0000 - val_fp: 193.0000 - val_tn: 402663.0000 - val_fn: 215.0000 - val_accuracy: 0.9809 - val_precision: 0.9822
- val_recall: 0.9803 - val_auc: 0.9993
```

Figure 07. The informative history result from the training phase.

2.5. After tuning

This part is about fine-tuning our pre-trained model, which means training our pre-existing model on its new tasks by adjusting the parameters, and then a new model's summary is printed.

II. RESULTS AND DICUSSION OF THE MODEL

1. Results

1.1 Preprocessing and augmentation results:

We took several steps during data preprocessing and augmentation to clean up the dataset and retain only information relevant to learning. Here are the results:

```
Number of validation batches: 272
Number of test batches: 68
```

Figure 08. Numbers of the test batches and Validation batches after the preprocessing.

The data augmentation is demonstrated on a sample image from the training dataset, and the augmented images are displayed as shown in the figure below.

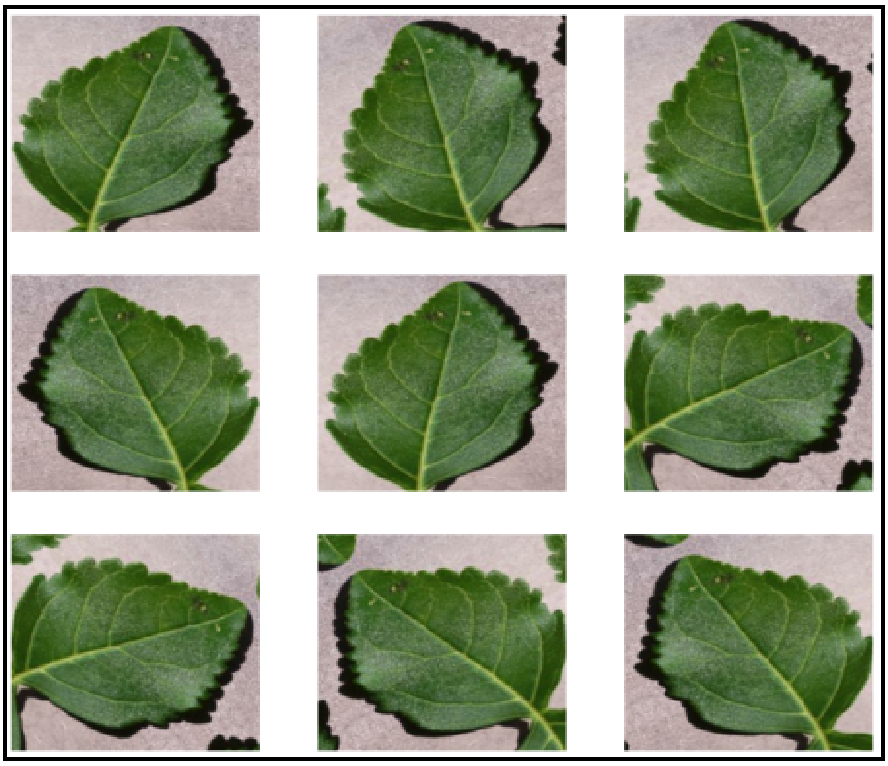

Figure 09. Augmentation of a sample image from the training dataset.

1.2. Training and validation accuracy graph:

After the training, the model was evaluated to check its effectiveness using a training and validation accuracy graph. The training and validation accuracy graph provides insights into the model's performance during the training process; it helps in determining how effectively our model generalizes to new, untested data. A model that can effectively identify pictures from the validation set and has a high validation accuracy has learnt relevant features. A low validation accuracy, on the other hand, indicates that the model may have trouble generalizing and may not function well with new data.

The training and validation accuracy graph of our model is presented in the **Figure 09**, it has three curves the training accuracy curve, this one represents the accuracy of the model on the training data throughout the training process. It shows how well the model is learning and improving its predictions as it iterates over the training dataset.

The validation accuracy curve that shows the model's accuracy on a different validation dataset that was not utilized for training. It evaluates the model's capacity for generalization and the precision of its forecasts for unknown data, and finally the start after tuning which represents the accuracy of the model after the tuning process.

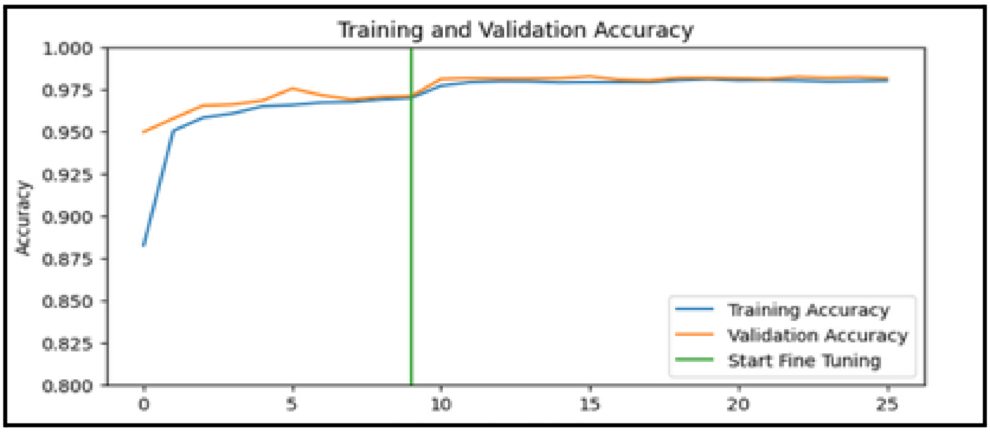

Figure 10. The training and validation accuracy graph.

1.3. The training and validation loss graph:

This graph is represented by the training, validation and start fine tuning curves; it visualizes the loss values that are experienced during the model’s training, validation and after tuning.

The training Loss curve displays the value of the loss function on the training data as the model passes through training iterations, while the validation loss curve indicates the loss on a different, non-training validation dataset and the start fine tuning curve evaluates the impact of the tuning procedure on the model's loss by reflecting the loss values of the model on the training or validation data after certain alterations or adjustments have been done.

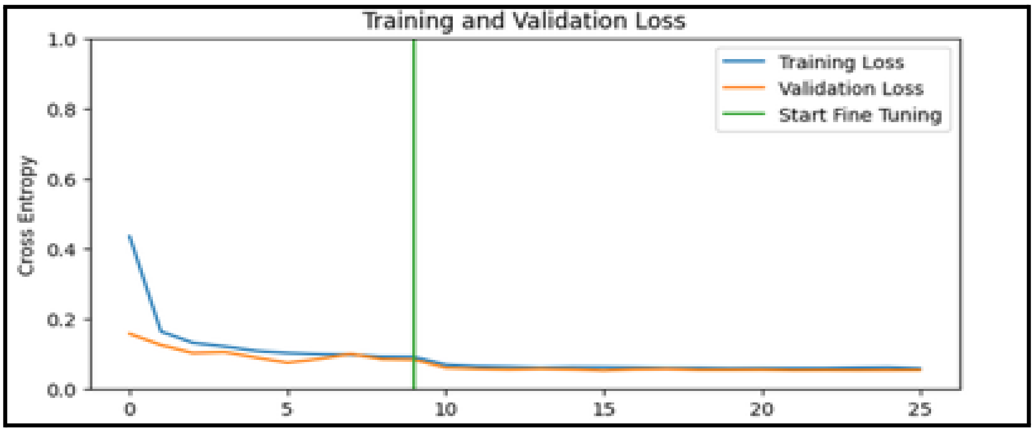

Figure 11. The training and validation loss graph.

1.4. Precision, accuracy, loss and Auc:

The performance of the model in terms of accuracy, false positives, false negatives, and discriminating ability may be better understood by considering the parameters above.

The performance curves of these parameters are depicted in **(Figure 12)**, showcasing their trends and changes over the course of the model's training and validation process. Additionally, **(Table 3)** provides the specific values of these parameters, allowing for a quantitative comparison and evaluation.

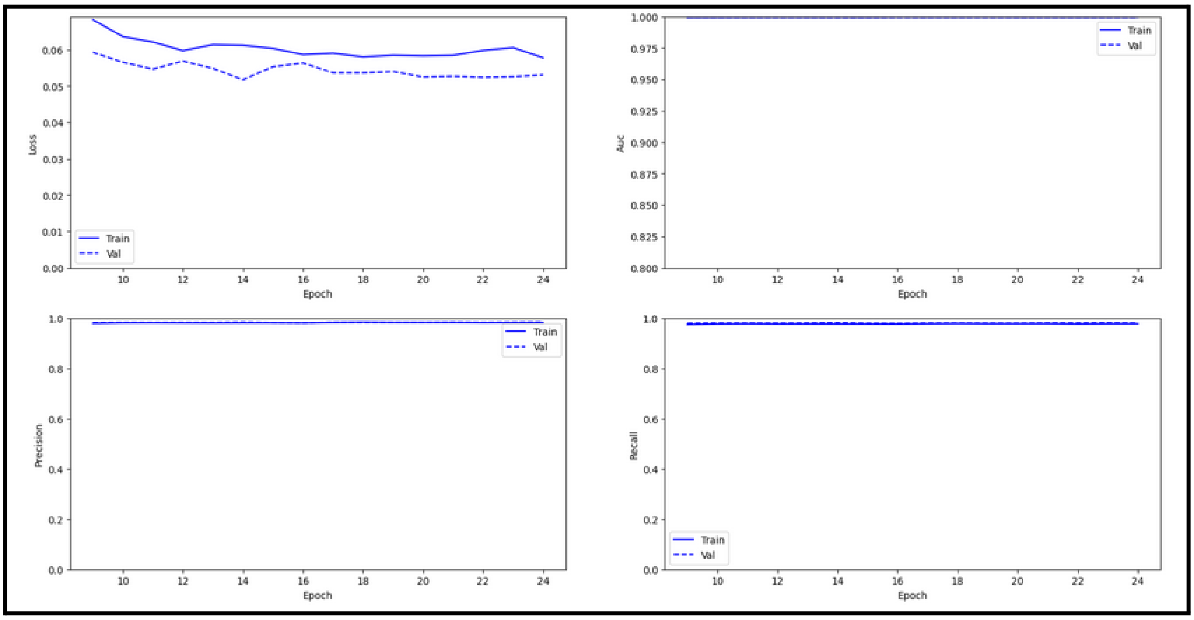

Figure 12. Graphic representations of the parameters: Loss, Precision, Recall and Auc.

Table 3. The values of the used parameters.

Parameter	Value
Accuracy	0.98
Precision	0.983
Recall	0.979
Auc	1.0
Loss	0.061

After evaluating our ResNet model, which was inspired by an existing model created by SACHINABHIMAY in 2021 and found on the Kaggle website, we integrated it into our Django website. As a result, we now have a fully functional website that is capable of performing diagnoses.

2. Discussion

The results presented in the previous graphs and table showcase the performance of our model in predicting crop diseases using the Plantvillage dataset. After training for 40 epochs, our model achieved remarkable accuracy and precision values. In the final epoch, the model demonstrated an accuracy of 97.48% and 98.22% on the training and validation data, respectively, along with a low loss value of 0.0826 and 0.0621.

These outcomes are very encouraging and demonstrate that our methods have allowed us to achieve a high level of correctness in predictions, and to assess the effectiveness of the model in minimizing errors. The ability of our model to precisely predict crop diseases shows the significance of our work in the field of phytopathogen prediction. By using the Plantvillage dataset, which comprises 38 classes of plant diseases that have been previously utilized in similar studies, we were able to capitalize on the extensive data available for training and testing purposes.

The utilization of the resnet-152 architecture participated to the success of our model. Resnet-152 is a deep neural network architecture known for its ability to handle complex image recognition tasks , this, played a crucial role in achieving the high accuracy and precision values observed in our results.

Overall, the obtained results highlight the model’s potential to assist farmers and agricultural professionals in making informed decisions related to disease control and management.

Chapter 4 :
Platform development

Platform development

I. Analysis and design

1. Agrilink Overview

Our platform is a crop diagnosis tool that aims to involve the use of artificial intelligence to improve the capacity of agricultural production.

The fact that farmers don't predict diseases of their crops at early stage causes them a huge losses, for this purpose the website is designed for farmers to help diagnose crop diseases and provide the suitable treatments by creating a link between farmers , providers of the used products , and the experts in-between. What shows effective results in decreasing the knowledge gap between providers and farmers by providing personalized advice and support.

The framework has a modular architecture that provides APIs that allow an easy integration of features, such as symptom recognition, data analysis and diagnosis engine.

2. Platform different features

2.1. Home page:

This is the front page of the website and the first page the user can see after reaching to the website, it provides an overview of the framework features and services, it includes the logo and the navigation menu that leads to the other different pages of the website.

The navigation menu contains five options, Log in/Sign in, About us, Farmer space, Provider space and the Expert space.

2.2 Login page:

This page provides the options of authentification, which is a crucial feature; the process of verifying the identity of a user of our website helps us protect the privacy of user data and allows users to manage their accounts. Moreover, authentification gives users the chance of having personalized advice and recommendations based on previous uses of the website.

2.3. About us page:

This option provides more information about our crop disease detection website; it displays the name and the logo, our purpose and values, our team members, our collaborations and our contact information.

2.4. Farmers space:

This option provides a space for farmers, it can be characterized as the Diagnosis Tool, and the page contains an image field where the farmer can upload the image of the diseased leaf to be analyzed and a text field where he can ask for advice or write a description. After sending the data, the user will get a response, which will be in another page, the result page.

These farmer's features are shown in **Figure 13** as follows:

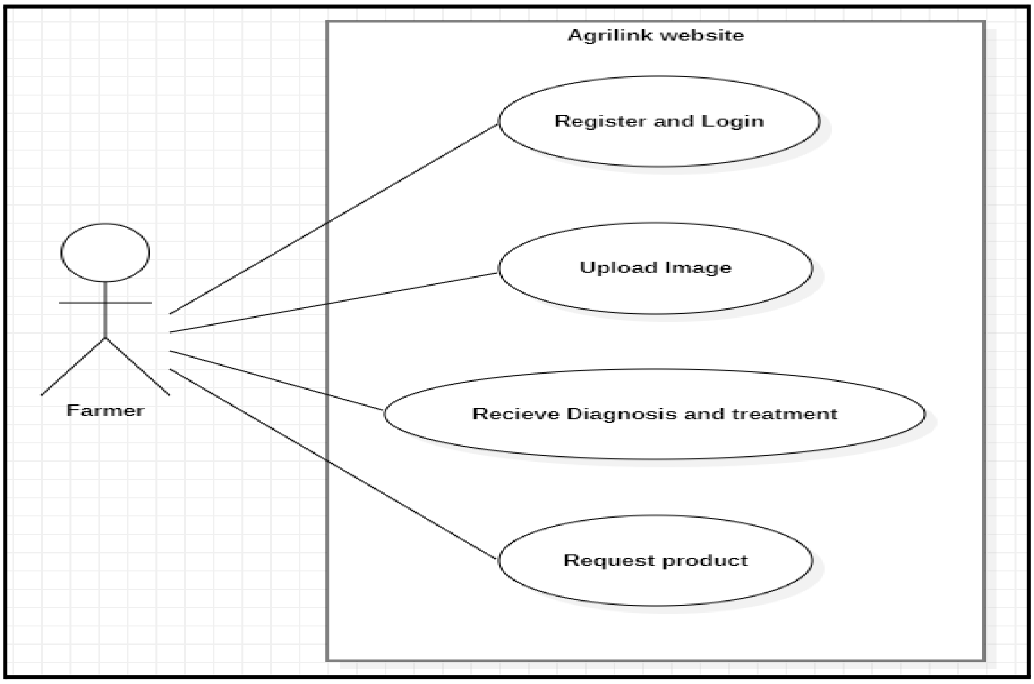

Figure 13. Features of the farmer in Agrilink platform.

2.5. Providers space:

The provider area is designed to support providers manage their business effectively and provide outstanding customer service to their farmer clients. In addition, this page contains features that allow the provider to manage his sales, such as the product management feature, it lets the providers manage their product listing like adding new products, updating prices and deleting products that are no longer available as it is shown in the figure below**(Figure 14).**

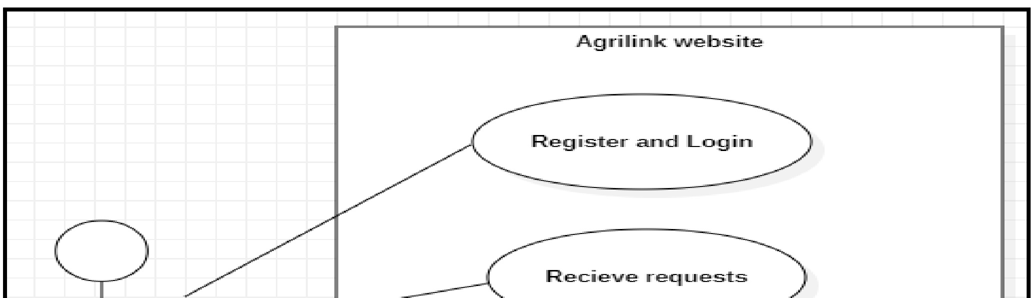

Figure 14. Features of the provider in Agrilink platform.

2.6. Experts space:

This page will connect farmers with knowledgeable experts, which will allow each farmer to have a personalized advice and support. It contains a messaging system to communicate with farmers who are seeking advice **(Figure15)**.

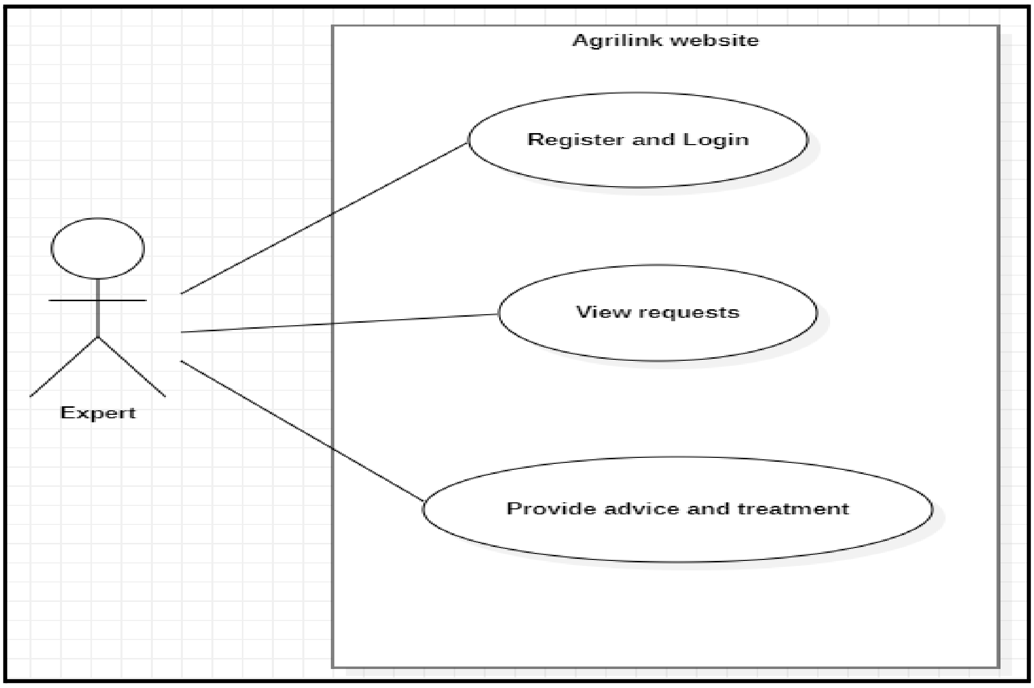

Figure 15. Features of the expert in Agrilink website.

2.7. Database tables list

Table	Action	Lignes	Type	Interclassement	Taille	Perte
auth_group	Parcourir Structure Rechercher Insérer Vider Supprimer	0	MyISAM	latin1_swedish_ci	1,0 kio	-
auth_group_permissions	Parcourir Structure Rechercher Insérer Vider Supprimer	0	MyISAM	latin1_swedish_ci	1,0 kio	-
auth_permission	Parcourir Structure Rechercher Insérer Vider Supprimer	36	MyISAM	latin1_swedish_ci	5,6 kio	-
auth_user	Parcourir Structure Rechercher Insérer Vider Supprimer	2	MyISAM	latin1_swedish_ci	3,3 kio	-
auth_user_groups	Parcourir Structure Rechercher Insérer Vider Supprimer	0	MyISAM	latin1_swedish_ci	1,0 kio	-
auth_user_user_permissions	Parcourir Structure Rechercher Insérer Vider Supprimer	0	MyISAM	latin1_swedish_ci	1,0 kio	-
customer	Parcourir Structure Rechercher Insérer Vider Supprimer	6	MyISAM	latin1_swedish_ci	2,5 kio	-
diagnostic	Parcourir Structure Rechercher Insérer Vider Supprimer	3	MyISAM	latin1_swedish_ci	2,4 kio	-
django_admin_log	Parcourir Structure Rechercher Insérer Vider Supprimer	13	MyISAM	latin1_swedish_ci	4,8 kio	-
django_content_type	Parcourir Structure Rechercher Insérer Vider Supprimer	9	MyISAM	latin1_swedish_ci	3,2 kio	-
django_migrations	Parcourir Structure Rechercher Insérer Vider Supprimer	32	MyISAM	latin1_swedish_ci	3,7 kio	-
django_session	Parcourir Structure Rechercher Insérer Vider Supprimer	2	MyISAM	latin1_swedish_ci	3,5 kio	-
product	Parcourir Structure Rechercher Insérer Vider Supprimer	0	MyISAM	latin1_swedish_ci	2,0 kio	44 o
13 tables	**Somme**	**103**	**MyISAM**	**latin1_swedish_ci**	**35,1 kio**	**44 o**

Figure 16. Database tables of the platform Agrilink.

3. Django architecture:

Developing an image-based plant disease diagnosis website using Django entails several crucial steps, including database setup, model creation, view implementation, and template integration. To ensure a professional and accurate approach, here are the necessary steps we followed:

- Setting up a Django project and creating a new app: Installing Django is the first step, which can be accomplished by executing the appropriate command in the command prompt. Once Django is installed, the next step is creating a new project using the Django framework. This will establish a new directory, with the fundamental structure required for a Django project.
- Creating a new app: Within a Django project, multiple apps can be developed. By creating a new app, a dedicated directory for the application is generated, including the essential structure for a Django app.

The steps provided above serve as a high-level prototype for constructing a web application based on image diagnosis, with the initial focus on outlining the Django-based structure.

Every Django website has three main code sections: input logic, business logic, and user interface logic and has a model-view-template (MVT) architecture that can be explained as follows:

3.1. Model: The model component assumes the responsibility of managing all tasks related to data manipulation and processing, when a Django model is created, a corresponding database table is automatically generated, this table reflects the structure of the model, with each attribute of the model being transformed into a field within the table.

The flow of information is as follows: when you sign up on any website, you click on a sign up button. When you click on the sign up button, a request is sent to the controller.

In Agrilink, we have implemented a user model that incorporates essential personal information of the user. This includes fields such as username, email, password, phone number, and role. The user model serves as a representation of the user entity and enables the storage and retrieval of user-specific data within the system; the models of our website reside in a 'models.py' file that is located within the individual app directory.

3.2. View: In Django's MVT (Model-View-Template) architecture, the view constitutes the second layer. Its primary role is to handle incoming web requests and generate suitable responses, which may involve rendering templates. The View is implemented as a Python function and is typically stored within the views.py file of the Django project, so in our website, when a user, such as a farmer, accesses our website, our goal is to direct them to a personalized home page designed specifically for farmers. This home page has been designed using an HTML template. To achieve this, we have implemented a corresponding view that handles the user's request, and responds by rendering the HTML template for the farmer's home page. As a result, whenever a farmer visits or requests the URL "http://127.0.0.1:8000/farmer/", they will be automatically presented with the farmer's home page, and this is applicable to handle all the user's requests such as uploading images, displaying diagnosis results and adding or deleting products.

3.3. Template: Templates in Django serve as a means to dynamically generate HTML for the project. Each Django template is typically denoted with the .html extension and comprises a blend of static content and dynamic elements. Django templates possess a distinctive syntax that introduces new methods for defining variables and tags within an HTML document.

In order to incorporate templates into our precision agriculture website, we established a dedicated directory within the project structure. This new directory, named "templates," serves as the central location for storing all HTML documents related to the website.

Examples of such HTML files include "provider.html," "homepage.html," "farmer.html," and various others.

Besides, after creating the templates, it is essential to ensure their visibility to Django by adding the templates directory to the TEMPLATES section of the settings.py file. This involves including the path of the templates directory within the appropriate configuration in the settings.py file, allowing Django to locate and utilize the templates effectively.

3.4. Urls: The urls.py file is located within the app directory in Django project, it is the main center for URL routing and mapping in a Django web application and it is in charge of establishing connections between incoming requests and the corresponding views that will handle those requests. Therefore, in our website we determine URL patterns using regular expressions or path patterns in the urls.py file, along with the corresponding views, so when a user visits a specific URL, Django checks the URL patterns defined in the urls.py file to determine which view or handler should be implemented to handle the request.

3.5. Settings: The settings.py file in a Django project serves as a central configuration file that contains various settings and parameters for the web application:

- Databases: Configuration settings for connecting to the database, which is MySQL in our website.
- Installed_apps: In this section, all the apps that are used in our Django project are mentioned, that are in our website “Diagnostic”,”Products”and “User”.
- Static_urls: The URL prefix for static files like CSS, JavaScript, and images.
- Language_code: The default language used in the application.
- Time_zone: The default time zone for date and time operations.
- Authentification_backends: A list of authentication backends to use.
- Templates: Configuration settings for handling templates [39].

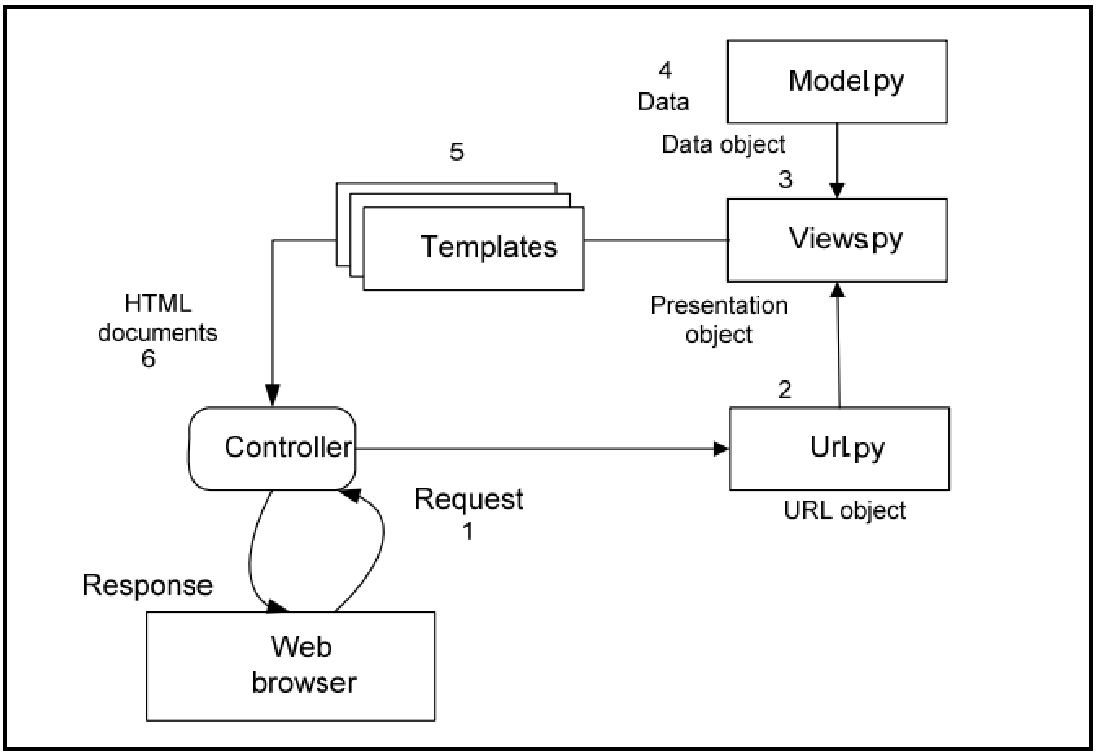

Figure 17. The architecture of a Django web application. [24]

The Figure **(Figure 17)** above presents an overview of the overall procedure for displaying a web page. As mentioned above; firstly, the Controller component receives a user request via a web browser. Subsequently, the Controller searches for a matching URL and executes the associated function of the view.py. In this process, the views.py file defines the template to be utilized for constructing the web page. It calls the relevant model function, which retrieves the necessary data. The returned data is then appropriately rendered and forwarded to the template. By combining the chosen template and the provided data, a web page is generated. Finally, as a result of the user's request, the HTML page is displayed in the web browser. [24]

II. Implementation

1. Languages and tools used to build our Django based platform:

- Python (3.9): is it already defined in the previous chapter.
- Django[7] : Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes

[7] https://Djangoproject.com/

care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source.

- HTML[8] : Stands for Hypertext Markup Language. It is a standard markup language for web page creation. It allows the creation and structure of sections, paragraphs, and links using HTML elements such as tags and attributes.
- CSS [9]: Stands for Cascading Style Sheets language and it is used to stylize elements written in a markup language such as HTML. It separates the content from the visual representation of the site.
- MySQL[10] : MySQL is an Oracle-backed open source relational database management system based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.
- Visual code studio[11]: Visual Studio Code is a free source -code editor developed by Microsoft. It is redefined and optimized for building and debugging modern web applications.

2. Code listings of our Django based website:

As previously mentioned, Django is an MVC (Model-View-Controller) framework. Here are some code listings showcasing functions from our Django website's views and models files :

[8] https://www.Developer.mozilla.org/
[9] https://www.Developer.mozilla.org/
[10] https://www.mysql.com/fr/
[11] https://code.visualstudio.com/

```python
class Diagnostic(models.Model):
    id = models.AutoField(primary_key=True)
    author = models.TextField(max_length=100, default="")
    title = models.TextField(max_length=200)
    description = models.TextField(2000)
    image = models.TextField(100)
    prediction = models.TextField(max_length=300, default="pas de predicition")
    reponse = models.TextField(max_length=1000, default="")


    def __str__(self):
        return self.title

    class Meta:
        db_table = 'diagnostic'
```

Figure 18. Diagnostic Model (models.py)

The code in **(Figure 10)** represents the Diagnostic model defined in the models.py file. The Diagnostic model inherits from Django's models.Model class and includes several fields such as id, author, title, description, image, prediction, and response. The code also shows that the Diagnostic model represents a database table with various fields to store information related to diagnostics.

Another model, which is the Product model **(Figure 11)**, represents a database table with fields to store information about products, including the provider, title, and price.

```python
class Product(models.Model):
    id = models.AutoField(primary_key=True)
    provider = models.TextField(max_length=50)
    title = models.TextField(max_length=50)
    price = models.TextField(max_length=50)


    def __str__(self):
        return self.title

    class Meta:
        db_table = 'product'
```

Figure 19. Product model (models.py)

```python
def FarmerUpload(request):
    import numpy as np
    if request.method == "POST":
        diag = Diagnostic()
        diag.title = request.POST['title']
        diag.description = request.POST['description']

        image_file = request.FILES['image']
        diag.image = image_file.name
        diag.author = request.session['username']
        print(image_file.name)

        image_path = os.path.join(settings.BASE_DIR, 'public/farmer', image_file.name)
        with open(image_path, 'wb') as file:
            for chunk in image_file.chunks():
                file.write(chunk)
        diag.save()

        return redirect('/user')
```

Figure 20. The FarmerUpload Function (views.py).

The **figure 20** represents the **FarmerUpload** function defined in the views.py file. It handles a POST request for uploading a diagnostic. The function creates a new instance of the Diagnostic model (diag) and assigns the title and description values from the POST data to the corresponding fields. It retrieves the uploaded image file and assigns its name to the image field of diag. The function then saves the image file and the diag instance to persist the diagnostic data in the database. Overall, it manages the process of uploading a diagnostic by creating an instance, saving the image file, and storing the diagnostic data.

```python
def InsertProduct(request):
    product = Product()
    product.title = request.POST['title']
    product.price = request.POST['price']
    product.provider = request.session['username']
    product.save()
    return redirect('/user')


def DeleteProduct(request):
    Product.objects.filter(id=request.POST['id']).delete()
    return redirect('/user')
```

Figure 21. The InsertProduct and DeleteProduct Functions (views.py).

In the Figure above, The **InsertProduct** function inserts a new product, while the **DeleteProduct** function removes a product based on its ID; both functions follow a similar principle, where they interact with the Product model to perform database operations.

The **InsertProduct** function handles a request to insert a new product into the database.

It creates an instance of the Product model and assigns the title and price values from the POST data. The provider field is assigned the value of the request's username session, indicating the provider's username. The product is then saved in the database, and the function redirects the provider to the '/user' URL.

While The **DeleteProduct** function handles a request to delete a product from the database. It retrieves the product with the specified ID and deletes it from the database. After deletion, the function redirects the user to the '/user' URL.

3. Visual Representation of the platform different pages:

- The home page:

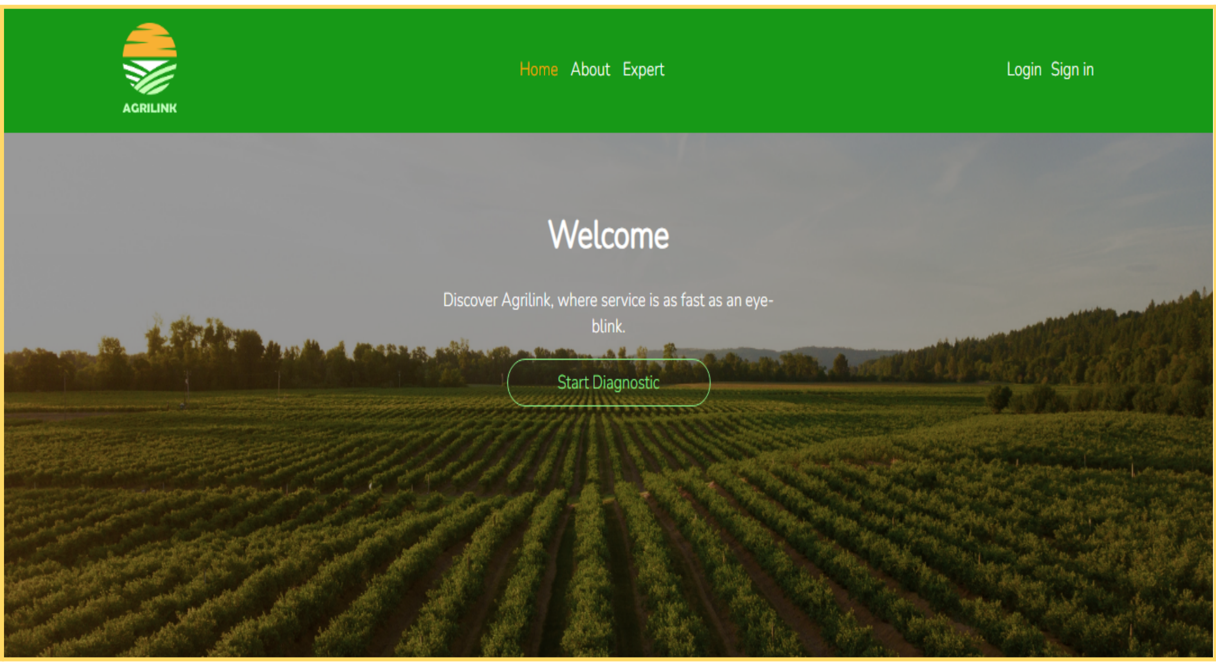

Figure 22. Visual representation of the Home page of the platform Agrilink.

- The Sign in Page :

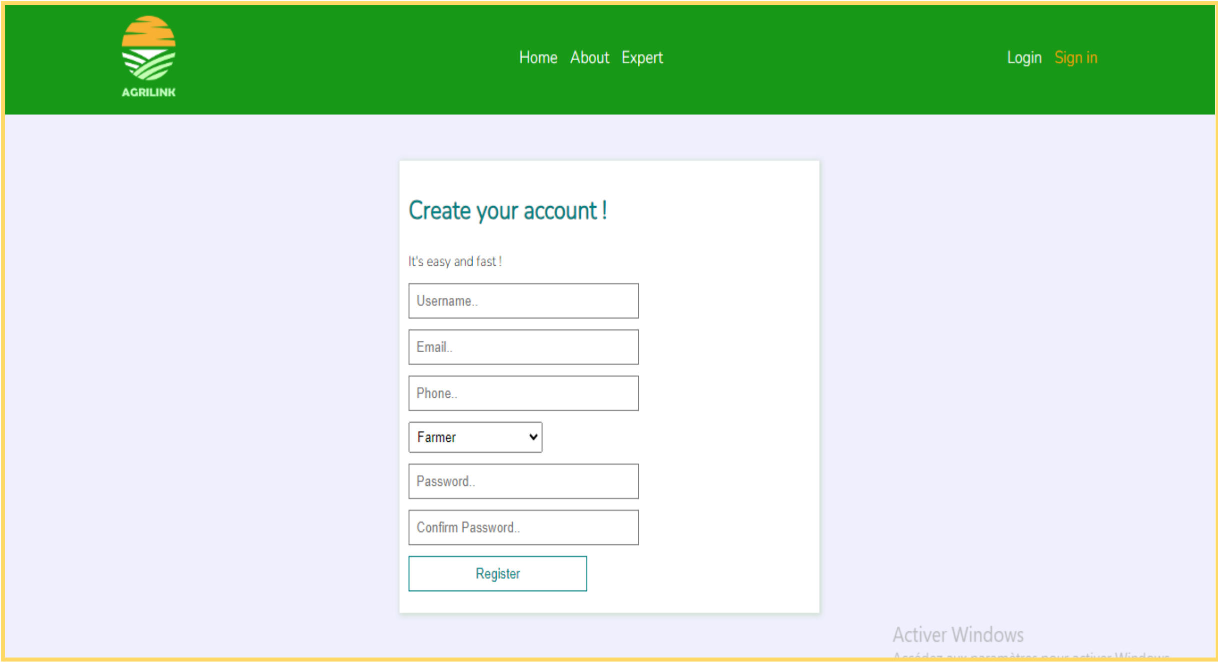

Figure 23. Visual representation of the Sign in page of the platform Agrilink.

- The Log in page:

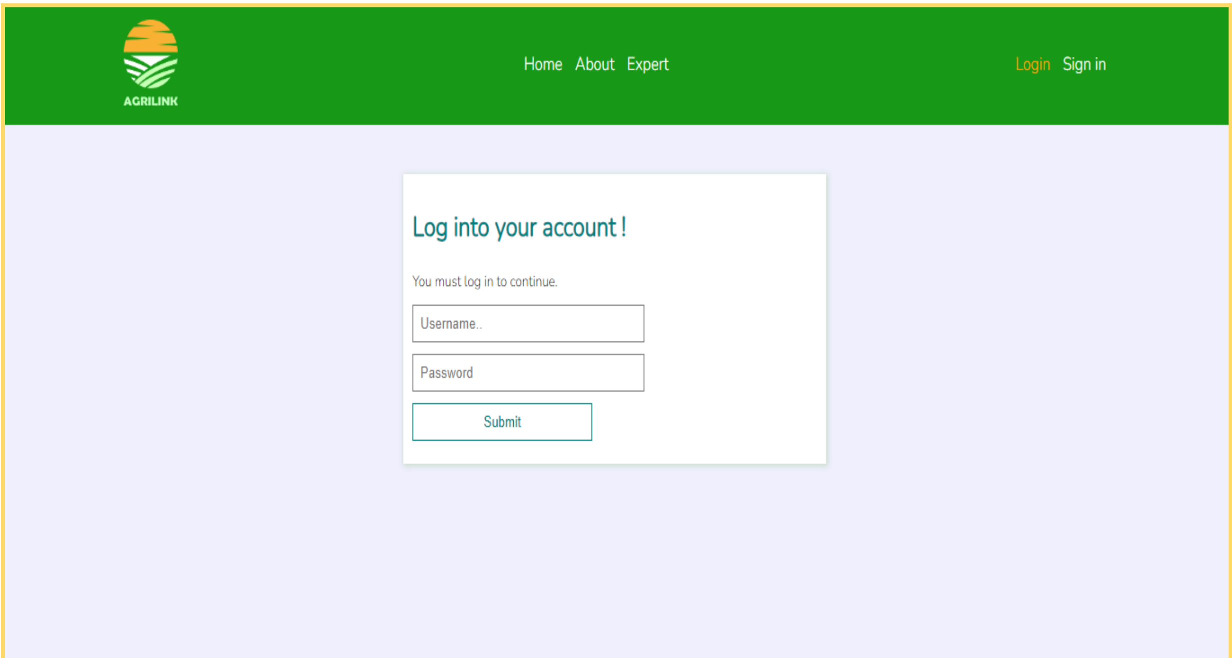

Figure 24. Visual representation of the Log in page of the platform Agrilink.

- The farmer’s page:

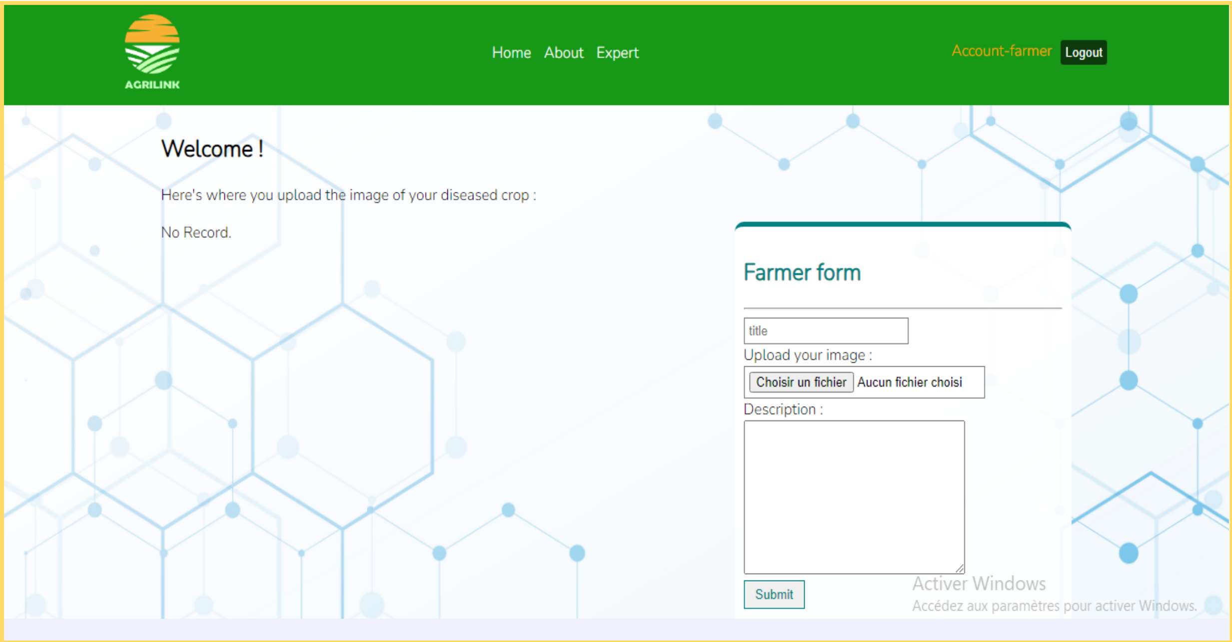

Figure 25. Visual representation of the Farmer’s page of the platform Agrilink.

- The provider’s page:

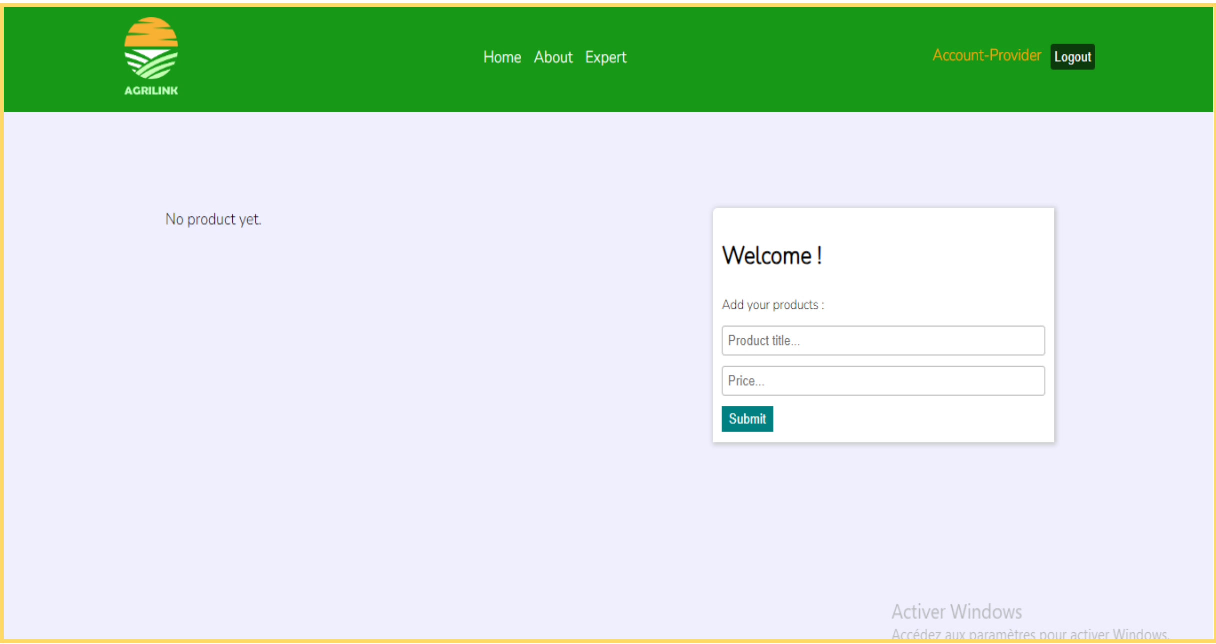

Figure 26. Visual representation of the provider’s page of the platform Agrilink.

- The expert’s page:

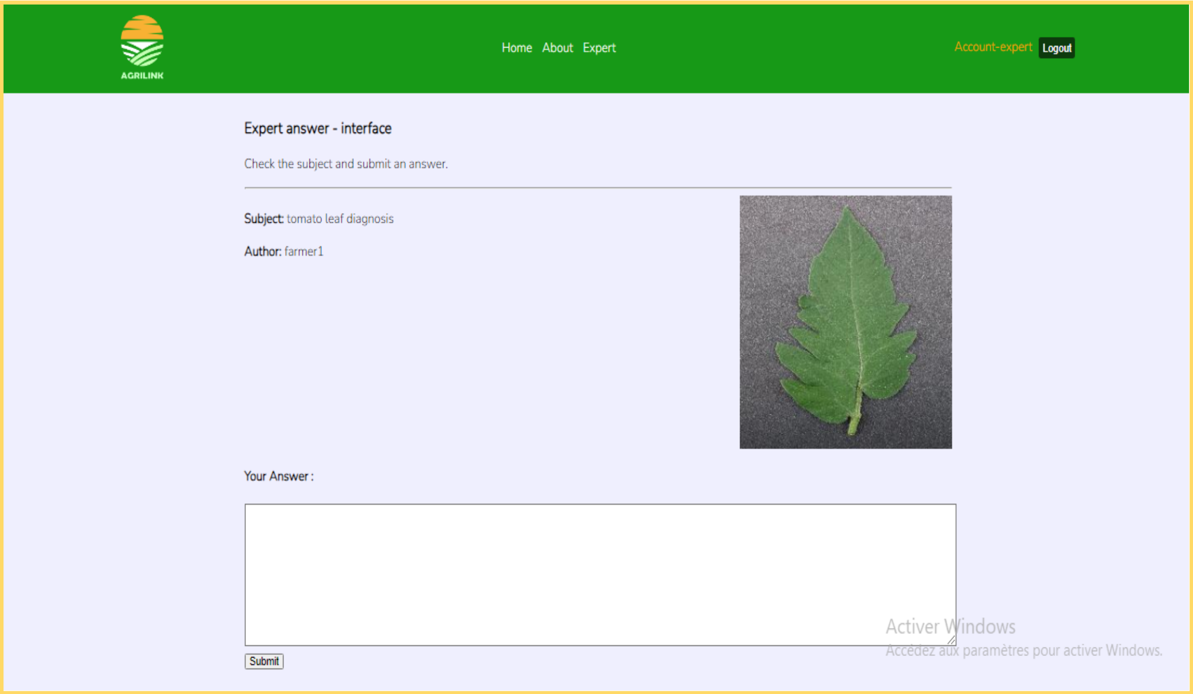

Figure 27. Visual representation of the expert's page of the platform Agrilink.

III. Deployment

After completing and finalizing our platform Agrilink, we deployed it on our university's server to make it accessible for different users. It may be accessed right now at this address: http://apps.umc.edu.dz:13254/.By employing the university's server, we guarantee reliable hosting and accessibility for customers who may now access the platform through this URL.

With this deployment, numerous agricultural stakeholders may use the platform with ease.

Conclusion

Conclusion

In conclusion, this dissertation has presented an artificial intelligence-based approach for phytopathogen identification through the use of a convolutional neural network (CNN) model combined with the design of a Django-based platform that facilitates diagnosis. The results obtained have demonstrated high accuracy on both the training and testing data, indicating the model's ability to generalize and make accurate predictions. This suggests that the proposed approach can be successfully used in broader studies and for phytopathogen detection, as well as in the functioning of our Agrilink platform, which allows farmers to have their own spaces where they can receive monitoring and advice from experts.

It is worth noting that similar works have used the same dataset as ours, which is the Plantvillage dataset, while others have utilized different datasets and architectures such as Inception_V3. Our work has shown a higher accuracy compared to other architectures. By combining the knowledge gained from these similar works, it is possible to further enhance the proposed approach by exploring new model architectures, techniques, and adding additional features to our platform.

To summarize, the use of artificial intelligence for phytopathogen detection through our platform offers a promising method that can open new perspectives in precision agriculture and holds great promise for transforming the industry.

As part of our future work, we aim to :

- Enhance our platform Agrilink.
- Introduce new functionalities.
- Improve the smart model by training it with Algerian data.
- Develop a model for sorting supplier results.
- Create a mobile app for farmers.

References

References

[1] Harris, D. R. (2020). Agriculture: Definition and Overview. In Encyclopedia of Global Archaeology, pp.140.

[2] Ludmila, P., Renata, A., & Jan J. (2020) Economic Aspects of Precision Agriculture Systems, Agris on-line Papers in Economics and Informatics, Volume XII, p.60.

[3] Robin, G., & Viacheslav, A. (2010) Precision Agriculture and Food Security, American Association for the Advancement of Science, Vol. 327, No. 5967, pp. 828-831.

[4] Babak, T., Ufuk, T., & Uğur Y. (2015) The Role of Precision Agriculture in the Promotion of Food Security, International Journal of Agricultural and Food Research, Vol. 4 No. 1, pp. 1-23.

[5] Gardner, B. L., & Rausser, G. C. (Eds.). (2002). Handbook of Agricultural Economics (first Ed.). Elsevier Science.

[6] Mrisho, L. M., Mbilinyi, N. A., Ndalahwa, M., Ramcharan, A. M., Kehs, A. K., McCloskey, P. C., Murithi, H., Hughes, D. P., & Legg, J. P. (2020). Accuracy of a Smartphone-Based Object Detection Model, PlantVillage Nuru, in Identifying the Foliar Symptoms of the Viral Diseases of Cassava-CMD and CBSD. *Frontiers in plant science*, *11*, 590889.

[7] Ahmed AA, Reddy GH. A. (2021) Mobile-Based System for Detecting Plant Leaf Diseases Using Deep Learning. AgriEngineering. 3(3):478-493.

[8] Chen, W.; Lin, Y.; Ng, F.; Liu, C.; Lin, Y. (2020) Ricetalk: Rice blast detection using internet of things and artificial intelligence technologies. IEEE Internet Things J.

[9] Jiang, P.; Chen, Y.; Liu, B.; He, D.; Liang, C.(2019) Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. 7, 69–80.

[10] Plantix. The Smart Crop Assistant on Your Smartphone. https://plantix.net/.

[11] Reddy, S.; Pawar, A.; Rasane, S.; Kadam, S. (2015) A Survey on Crop Disease Detection and Prevention using Android Application. Int. J. Innov. Sci. Eng. Technol., 2, 621–626.

[12] Agrotic Chair. (2018). Deep learning and agriculture. (Chair Agrotic, Trans).

[13] Arne Wolfewicz (June 2022). 18 mars 2023.
https://levity.ai/blog/difference-machine-learning-deep-learning.

[14] Saumyab217. (2021, June 14). Machine Learning vs Deep Learning vs Artificial Intelligence| Know in-depth Difference [JPG]. Analytics vidhya. Retrieved March 18, 2023, https://www.analyticsvidhya.com/blog/2021/06/machine-learning-vs-artificial-intelligence-vs-deep-learning/ .

[15] Riedl, M. O. (2019). Human‐centered artificial intelligence and machine learning. Human Behavior and Emerging Technologies, e117.

[16] TechTarget. (n.d.). Machine Learning (ML) definition. Retrieved from https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML.

[17] Howers, T., Houtenbos, J., Ter Velde, K., & Alderding, B. (2023). Farm21. Retrieved from https://www.farm21.com/fr/.

[18] Linkinfarm (2017).https://www.linkin.farm/ .

[19] Yash, D., Piyush, D., Umesh, J., & Sanober, S. S. (2020). Analysis of different CNN models for plant disease detection and identification. International Research Journal of Engineering and Technology (IRJET), 7(5), 2685.

[20] Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. Insights into Imaging, 9, 611-629.

[21] Etuk, E. A., & Ayuk, J. O. (2021). Agricultural commercialisation, poverty reduction and pro-poor growth: evidence from commercial agricultural development project in Nigeria. Helyion, 7.

[22] Sumberg, J., & Giller, K. (2022). What is 'conventional' agriculture? Global Food Security, 32, 4.

[23] Jool International. (2022, March 28). Agriculture Extensive VS Intensive. Retrieved from https://jool-international.com/agriculture-_-environnement_-cote-divoire/)

[24] Savić, D., Ilić, M., Sodnik, J., Kos, A., Stancin, S., & Tomazic, S. (2008). Simulation data exchange - web interface for CostGlue application. Research Gate, 297264757, 4-5.

[25] Noyan, M. A. (2022). Uncovering bias in the PlantVillage dataset. arXiv e-prints, 2206.04374v1, 1-3

[26] Machine Learning in MATLAB. (n.d.). Retrieved from https://www.mathworks.com/help/stats/machine-learning-in-matlab.html

[27] AnalystPrep. (n.d.). Supervised Machine Learning, Unsupervised Machine Learning, Deep Learning. Retrieved from https://analystprep.com/study-notes/cfa-level-2/quantitative-method/supervised-machine-learning-unsupervised-machine-learning-deep-learning/

[28] Blanc, P. (2018). Réseaux de neurones convolutifs en médecine nucléaire : Applications à la segmentation automatique des tumeurs gliales et à la correction d'atténuation en TEP/IRM. Mémoire de recherche non publié, Université Paris Descartes, Faculté de Médecine Paris Descartes.

[29] The difference between supervised and unsupervised learning from

https://www.datagenius.fr/post/modeles-predictifs-et-automl-quelle-solution-d-intelligence-artificielle-choisir

[30] Sial, A., Shankar, T., Praharaj, S., Sahoo, U., & Maitra, S. (2021). Intensive Farming: Its Effect on the Environment. Indian Journal of Natural Sciences, 12(69), 37480.

[31] Teradata. (n.d.). What is Python? Retrieved May 27, 2023, from https://www.teradata.com/Glossary/What-is-Python.

[32] Django Software Foundation. (n.d.). Django web framework. Retrieved from https://www.djangoproject.com/.

[33] Troell, M., Naylor, R. L., Metian, M., Beveridge, M., Tyedmers, P. H., Folke, C., Arrow, K. J., Barrett, S., Crépin, A.-S., Ehrlich, P. R., Gren, Å., Kautsky, N., Levin, S. A., Nyborg, K., Österblom, H., Polasky, S., Scheffer, M., Walker, B. H., Xepapadeas, A., & de Zeeuw, A. (2019). Sustainable aquaculture: Progress, challenges, and opportunities. Science, 365(6455), eaaw3656.

[34] Lal, R. (2019). Sustainable Soil Management: Preventive and Ameliorative Strategies (Sustainable Agriculture Reviews, Vol. 29). Springer.

[35] Smith, L. G., Williams, A. G., & Pearce, B. D. (2015). The energy efficiency of organic agriculture: A review. Renewable Agriculture and Food Systems, 30(3), 280-301.

[36] Pure Greens. (2023). What Is Commercial Agriculture? Pure Greens. Retrieved from https://puregreensaz.com/what-is-commercial-agriculture/

[37] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[38] AIT ISSAD, H.(2020) Déploiement intelligent de drones pour une agriculture du futur, Université Mouloud Mammeri de Tizi-Ouzou.

[39] Django Architecture – Detailed Explanation Retrieved from https://www.interviewbit.com/blog/django-architecture/

Année universitaire :2022-2023	Presented by : Khelifi Djoumana Bentounsi Ines

AGRILINK : A platform for plant pathogens detection and treatment

Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of Master in Bioinformatics

The objective of this work is to address the challenges in crops management and phytopathogen detection by developing a platform that enables farmers to obtain quick and accurate diagnoses using artificial intelligence. By facilitating early detection and precise diagnosis of phytopathogens, our platform aims to reduce the reliance on chemical products in agriculture, contributing to sustainable farming practices.This platform also functions as a center that connects farmers with phytosanitary product providers and agricultural experts providing collaboration and knowledge sharing that will foster a sense of community among stakeholders in the agricultural sector. Through this platform, farmers can access expert advice and recommendations, share best practices, and make informed decisions about crop protection and management strategies.

Key words : Precision agriculture , pathogenes detection , artificial intelligence diagnostic

Chairman: Dr. M.S DAAS

Supervisor: Dr. H. CHEHILI

Examiner: Mr. K. KELLOU

Incubator: Pr. I. BELLIL

CATI: Dr. S. BETINA

Socio-economic partner: Mr. R. Bouarroudj, Agrichem Company