PEOPLE’S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research
Mohamed Khider University BISKRA
Faculty of Exact Sciences, Natural and Life Sciences

Computer Science Department

Order Number: Startup_08/RTIC/M2/2024

Thesis

Presented to obtain the academic master’s degree in
Computer Science

Option: **Networks and Information and Communication Technologies**

Title

A mechanism to detect and eliminate harmful insects in smart greenhouses

Presented by:	Supervised by:
-TENGOURI NADA	**-Dr.MERIZIG ABDELHAK**
-CHENNOUFI DOUNIA	**-Dr.AYAD SOHEYB**

Defended on June 24, 2024: In front of the jury composed of:

-TERRISSA Sadek Labib	Prof	President
-Merizig Abdelhak	MCB	Supervisor
-SAHLI Sihem	MAA	Examiner

Academic year : **2023/2024**

Acknowledgement

*First and foremost, we express our gratitude to **Allah** for blessing us with health, willpower, strength, courage, and patience. These qualities have enabled us to overcome challenges and achieve our goals, without which our project would not have reached fruition.*

*We extend our sincere thanks and appreciation to our thesis supervisors, **Dr. Merizig Abdelhak** and **Dr. Ayad Soheyb**, for their exceptional guidance, support, and instructions.*

*We also wish to thank the members of the jury,**Prof. TERRISSA Sadek Labib**, **Dr. SAHLI Sihem**, and **Dr. Merizig Abdelhak** , for their time, effort, and valuable feedback in reviewing our work.*

*We extend our sincere gratitude to all those who supported us throughout our academic journey, especially during this challenging year. We offer a special thanks and gratitude to **Dr. Bellaala Abir**, who has been a great help to us.*

*We are deeply thankful to our families, the **Chennoufi** and **Tengouri** families, for their inspiration, assistance, and the time they devoted to helping us make our dissertation shine.*

We also express our gratitude to our friends and colleagues for their moral and intellectual support throughout our work. We hope they recognize the impact of their support when reading these words.

Remerciements

Tout d'abord, nous exprimons notre gratitude à ***Allah*** *pour nous avoir bénis avec la santé, la volonté, la force, le courage et la patience. Ces qualités nous ont permis de surmonter des défis et d'atteindre nos objectifs, sans lesquels notre projet n'aurait pas abouti.*

Nous adressons nos sincères remerciements et notre appréciation à nos directeurs de thèse, ***Dr Merizig Abdelhak*** *et* ***Dr Ayad Soheyb****, pour leurs conseils, leur soutien et leurs instructions exceptionnels.*

Nous souhaitons également remercier les membres du jury, ***Prof. TERRISSA Sadek Labib****,* ***Dr. SAHLI Sihem****, et* ***Dr. Merizig Abdelhak*** *, pour leur temps, leurs efforts et leurs précieux commentaires dans l'examen de notre travail.*

Nous exprimons notre sincère gratitude à tous ceux qui nous ont soutenus tout au long de notre parcours académique, en particulier durant cette année difficile. Nous adressons un merci spécial et une profonde gratitude à ***Dr. Bellaala Abir****, qui nous a beaucoup aidés.*

Nous sommes profondément reconnaissants à nos familles, les familles ***Chennoufi*** *et* ***Tengouri****, pour leur inspiration, leur aide et le temps qu'elles ont consacré à nous aider à faire briller notre thèse.*

Nous exprimons également notre gratitude à nos amis et collègues pour leur soutien moral et intellectuel tout au long de notre travail. Nous espérons qu'ils reconnaîtront l'impact de leur soutien en lisant ces mots.

شكر وتقدير

أولاً وقبل كل شيء، نتوجه بخالص الشكر لله على نعمة الصحة، والإرادة، والقوة، والصبر. إن هذه الصفات النبيلة هي التي أمكنتنا تجاوز التحديات وتحقيق أحلامنا، ولولاها ما تحققت ثمار مشروعنا.

نود أن نعبر عن شكرنا العميق والمخلص لمشرفينا في الرسالة، الدكتور **مريزيق عبد الحق** والدكتور **عياد صهيب**، على توجيههما الرائع ودعمهما اللامحدود .

وكذلك، نتقدم بخالص الشكر لأعضاء اللجنة: البروفيسور تيريسا لبيب ، الدكتور مريزيق عبد الحق والدكتورة سهلي سهام على وقتهم وجهدهم وملاحظاتهم القيمة التي ساهمت في تطوير عملنا.

نشكر بعمق كل من دعمنا خلال رحلتنا الأكاديمية، ولاسيما خلال هذا العام الصعب. ونتوجه بشكر خاص وامتنان للدكتورة **بلعلى عبير** ، التي كانت لها يد كبيرة في مساعدتنا.

ونعبر عن امتناننا العميق لعائلتينا، عائلتي شنوفي وطنقوري، على دعمهم اللامحدود وتفانيهم في مساعدتنا على تأليف رسالتنا.

وأخيرًا، نعبر عن شكرنا العميق لصديقاتنا وزملائنا وزميلاتنا على دعمهم الروحي والفكري الذي لا يُقدَّر بثمن طوال فترة عملنا. ونأمل أن يدركوا تأثير دعمهم العظيم عند قراءتهم لهذه الكلمات البسيطة.

Abstract

Agriculture is one of the pillars of Algeria's economy, especially tomato cultivation, which is increasingly popular due to its various uses and benefits. Agriculture within plastic greenhouses is considered one of the common and important methods to meet the growing demand, as it provides a suitable environment. However, farmers face challenges from pests and harmful insects, especially Tuta absoluta, which traditional methods have not shown effectiveness in combating.

Tuta absoluta is considered one of the most dangerous pests affecting crops in plastic greenhouses, especially tomatoes, causing significant damage and high costs for farmers, leading to major economic losses.

To address this challenge, our project introduces an innovative idea proposing a new approach that combines modern technologies such as artificial intelligence and smart agriculture. We introduce a smart device that detects and efficiently eliminates this insect. The proposed device consists of sensors and a camera, where the camera operates automatically when the motion sensor detects the presence of this insect, transmitting real-time data to a deep learning model that accurately determines the adult stage of Tuta absoluta. The device is strategically placed near leds(yellow and green) and traps to attract, capture, and eliminate the insect. The system also includes a process to predict the emergence of this pest using machine learning models. Additionally, the project includes a mobile application that instantly informs farmers about detection, elimination, and prediction events, empowering them to make quick and effective decisions to protect their crops.

Through this innovative approach, the flexibility of artificial intelligence is leveraged simultaneously with the Internet of Things, leading to increased accuracy and effectiveness at a reasonable cost, to enhance the fight against Tuta absoluta.

Keywords Smart Agriculture(SA), tuta Absoluta, sensors , artificial intelligence (AI), machine learning(ML), deep learning(DL).

Résumé

L'agriculture est l'un des piliers de l'économie algérienne, notamment la culture des tomates qui bénéficie d'une popularité croissante en raison de ses multiples utilisations et avantages. L'agriculture sous serre plastique est considérée comme une méthode courante et importante pour répondre à la demande croissante, offrant un environnement propice. Cependant, les agriculteurs sont confrontés à des défis posés par les ravageurs, en particulier le ravageur Tuta absoluta, pour lequel les méthodes traditionnelles de lutte n'ont pas montré leur efficacité.

Le Tuta absoluta est l'un des ravageurs les plus dangereux pour les cultures sous serre plastique, en particulier les cultures de tomates, causant des dommages graves aux récoltes et des coûts très élevés pour les agriculteurs, entraînant d'importantes pertes économiques.

Pour relever ce défi, notre projet propose une idée novatrice qui propose une nouvelle approche combinant les technologies modernes telles que l'intelligence artificielle et l'agriculture intelligente. Nous proposons un dispositif intelligent qui détecte et élimine efficacement ce ravageur. Le dispositif proposé comprend des capteurs et une caméra, la caméra fonctionnant automatiquement lorsque le capteur de mouvement détecte la présence de ce ravageur, transférant ensuite les données en temps réel à un modèle d'apprentissage en profondeur qui détermine avec précision le stade adulte de Tuta absoluta. De plus, le dispositif est stratégiquement placé près des LED (jaune et vert) et des pièges pour attirer le ravageur, le capturer et l'éliminer. Le système comprend également un processus de prédiction de l'apparition de ce ravageur à l'aide d'un modèle d'apprentissage automatique. De plus, le projet comprend une application mobile qui informe instantanément les agriculteurs des événements de détection, d'élimination et de prédiction, leur permettant de prendre des décisions rapides et efficaces pour protéger leurs cultures.

Grâce à cette approche novatrice, la flexibilité de l'intelligence artificielle est exploitée en synchronisation avec l'Internet des objets, ce qui permet d'augmenter sa précision et son efficacité à un coût raisonnable, renforçant ainsi le processus de lutte contre la Tuta absoluta.

Mots-clés Agriculture Intelligente (AI), tuta Absoluta, capteurs, intelligence artificielle (IA), apprentissage automatique (ML), apprentissage profond (DL).

ملخص

الزراعة هي أحد أركان اقتصاد الجزائر، خصوصا زراعة الطماطم التي تحظى بشعبية متزايدة لاستعمالاتها وفوائدها المتعددة. وتُعتبر الزراعة داخل البيوت البلاستيكية من الأساليب الشائعة والمهمة لتلبية الطلب المتزايد حيث توفر بيئة ملائمة. ومع ذلك، يواجه المزارعون تحديات من قبل الحشرات والآفات الضارة وخصوصا حشرة توتا أبسولوتا التي لم تُظهِر الطرق التقليدية فعاليتها لمكافحة هذه الآفة.

تعتبر حشرة توتا أبسولوتا من اخطر الآفات التي تصيب المحاصيل في البيوت البلاستيكية وأولها محاصيل الطماطم حيث تسبب أضراراً جسيمة للمحصول وتكاليف جد مرتفعة للمزارعين، مما يؤدي إلى خسائر اقتصادية كبيرة.

لمواجهة هذا التحدي، يُقدم مشروعنا فكرة مبتكرة تقترح نهج جديد يجمع بين التقنيات الحديثة كالذكاء الاصطناعي والزراعة الذكية. حيث نقدم جهاز ذكي يكتشف هذه الحشرة ويقضي عليها بكفاءة. يتكون الجهاز المقترح من مستشعرات وكاميرا, حيث تشتغل الكاميرا تلقائيا عندما يستشعر مستشعر الحركة وجود هذه الحشرة والتي بدورها تنقل البيانات في الوقت الفعلي إلى نموذج التعلم العميق الذي يحدد بدقة مرحلة توتا ابسلوتة البالغة, وقد تم وضع الجهاز بشكل استراتيجي بالقرب من المصابيح (الأصفر والأخضر) و من المصائد ليقوم النظام بجذب الحشرة وعملية التقاطها والقضاء عليها. كما يتضمن النظام عملية توقع ظهور هذه الآفة باستخدام نموذج تعلم الآلة. بالإضافة إلى ذلك، يتضمن المشروع تطبيقًا للهواتف المحمولة يُعلم المزارعين بشكل فوري عن أحداث الكشف والقضاء والتنبؤ، مما يمنحهم القدرة على اتخاذ القرارات السريعة والفعّالة لحماية محاصيلهم.

ومنه ومن خلال هذا النهج المبتكر، يتم استغلال مرونة الذكاء الاصطناعي بالتزامن مع إنترنت الأشياء، مما يؤدي إلى زيادة دقته وفعاليته و بتكلفة معقولة، لتعزيز عملية مكافحة توتا أبسولوتا.

الكلمات المفتاحية:

الزراعة الذكية، توتا ابسولوتا، الاجهزة الاستشعارية، الذكاء الاصطناعي، التعلم الآلي، التعلم العميق.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Equations

Chapter 1

General Introduction

1.1 General Context

Tuta absoluta, also known as the tomato leaf miner, is a devastating pest that poses a significant threat to tomato crops worldwide. This insect infests tomato plants, causing extensive damage to leaves, stems, and fruits, leading to severe economic losses for farmers. Traditional pest control methods have often proven ineffective against Tuta absoluta due to its rapid reproduction rate and ability to develop resistance to pesticides.

In our current situation, technology assists us in various ways. A motion sensor detects Tuta absoluta movements and commands a camera to classify if it is an adult. Once identified, a aspirator initiates a targeted elimination process, reducing the Tuta absoluta population and minimizing crop damage. The DHT11 sensor measures temperature and humidity, and a machine learning model predicts the insect's presence, notifying the farmer via a mobile application to take necessary measures. All of this is achieved through artificial intelligence.

1.2 Problem statement

Regrettably, Algerian farmers continue to rely on conventional techniques for eradicating tomato leaf miners, presenting significant challenges such as:

- Ineffectiveness and environmental harm associated with traditional pest control methods.
- Excessive pesticide use, leading to environmental pollution and health hazards.
- Challenges in early detection and targeted elimination of Tuta absoluta larvae.
- Pesticide resistance development in Tuta absoluta.

1.3 Objectives

The goals of our project include :

- Develop a model for early detection and elimination of Tuta absoluta.
- Minimize environmental impact by reducing pesticide usage.
- Improve crop yield and quality through effective pest control.
- Ensure timely intervention to prevent widespread infestations.
- Integrate modern technologies for comprehensive pest management.

1.4 Thesis Structure

The current report is structured into Five chapters:

- **Chapter 01:**In this chapter, we will discuss general information about smart Agriculture
- **Chapter 02:** The seconde chapter contains some artificial intelligence concepts such as machine learning, deep learning, and image classification
- **Chapter 03:** The third chapter includes the design of our system and a number of diagrams explaining the functionality of our system.
- **Chapter 04 :**The fourth chapter presents a list of development tools and the programming language used for the development of our project, along with pseudo-algorithms illustrating the system's operations. We concluded it with images showcasing our practical application aspec
- **Chapter 05 :'Conclusion and Perspectives'** This concludes the main goal of this paper and it gives a point of view for future work

Chapter 2

Smart Agriculture Overview

2.1 Introduction

Smart Agriculture revolutionizes farming practices through the integration of advanced technologies. It is instrumental in tackling agricultural hurdles such as harmful insects.

This chapter provides a comprehensive overview of Smart Agriculture. Starting with its definition and the evolutionary trajectory of agriculture. Then, it discusses the paramount significance of Smart Agriculture in modern farming practices, highlighting the pivotal role played by Information and Communication Technologies (ICTs). After that, it offers insights into Greenhouse Farming, giving a succinct overview of greenhouse techniques and explaining the challenges posed by harmful insects, along with the various management strategies for these pests.

2.2 Definition of Smart Agriculture

Smart Farming integrates advanced Information and Communication Technologies (ICTs) like internet of thing (IoT) and Cloud Computing into agricultural practices, enabling precision farming. emphasizing these technologies are poised to introduce increased automation through robotics and AI. Big Data, with its vast, diverse datasets, plays a pivotal role, enabling informed decision-making in agriculture [13].

Precision farming has the potential to greatly enhance agricultural output in both productivity and sustainability aspects [14]. It considered a facet of smart agriculture that employs IoT technology to analyze field data for optimizing resource usage and improving productivity.Through sensors and data analysis, Farmers carefully consider their options when it comes to planting, fertilizing, and determining the best times for harvesting, and determining optimal harvest times. IoT devices aid in monitoring crop health, moisture levels, and remotely. By leveraging IoT, farmers can proactively address agricultural challenges, such as water scarcity, harmful insects detection and soil variability, leading to increased efficiency and cost-effectiveness in farming practices [15].

2.3 Evolution of Agriculture

Agriculture is the primary source of food and raw materials [16]. The diagram charts agriculture's progression through four major stages:

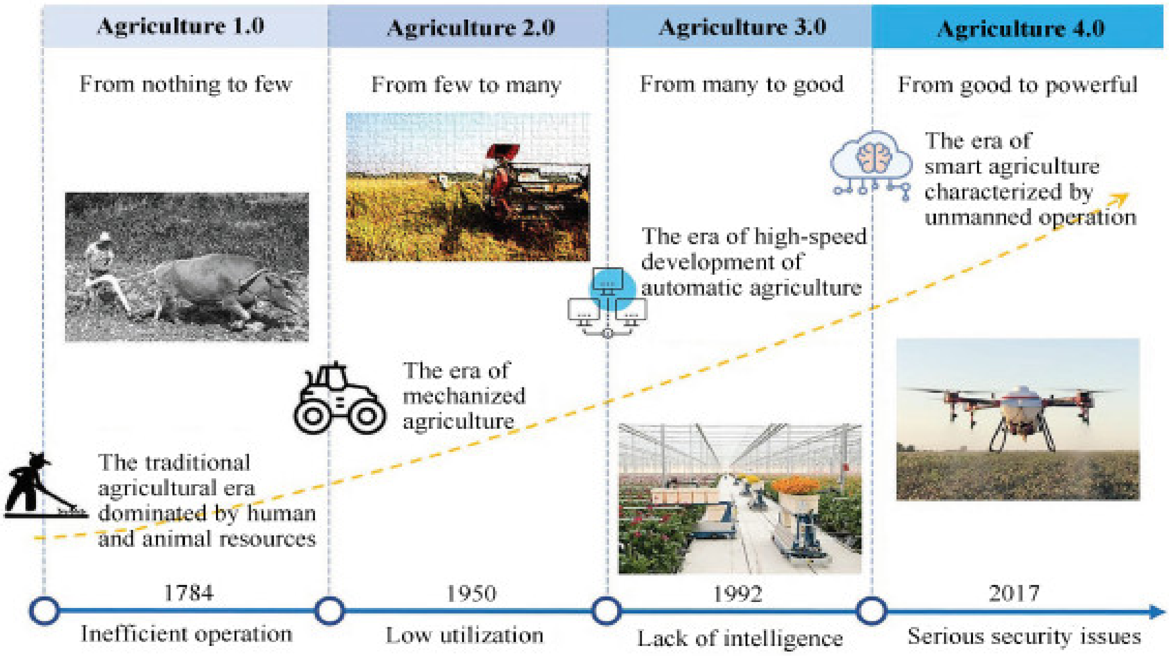


Figure 2.1: Evolution of Agriculture [1]

2.3.1 Agriculture 1.0: Traditional Farming

The Traditional Agriculture dominance was around 1784 [1].

- During this period, there was a strong emphasis on manual human labor and animal power [17]. Its advantages rest in the sustainable production of food via methods like horticulture, arboriculture, and vegeculture, which have been crucial for supplying food to different communities. Conventional farming methods have also been instrumental in shaping landscapes and contributing to the human diet by cultivating a diverse range of crops and livestock [18].
- Limitations of this phase:
  - ▶ Despite its potential, the absence of technology leaded to inefficient operations and low yields [1].
  - ▶ Low production yields due to dependence on physical exertion.
  - ▶ Susceptible to variations in weather conditions and pest infestations [19].

2.3.2 Agriculture 2.0: Mechanization

The second agricultural revolution around 1950 was a pivotal shift in farming practices [17] [1].

- The introduction of machinery like tractors revolutionized farming, boosting efficiency and crop yields, fostering food stability and economic growth [19].

- Additionally, reducing manual labor because of the machinery to automate tasks like plowing.
- This era saw the early use of synthetic fertilizers and pesticides.
- Limitation of this stage:
  - ▶ Environmental costs: increased pesticide use, soil erosion, water depletion, and salinization of irrigated lands, so it is waste of natural resources. [20].
  - ▶ Gender disparities: women faced health risks from pesticide exposure, deforestation for fuelwood collection, and water contamination.
  - ▶ Social inequities: unequal distribution of benefits, leading to marginalization of small-scale farmers and women.
  - ▶ High initial investment costs: posing a barrier to entry for small-scale farmers [21].

2.3.3 Agriculture 3.0: Precision Farming

The Agriculture 3.0, also known as Green Revolution or Precision Agriculture, emerged around 1992, heralding a new era of farming techniques [17] [1].

- Advancements in technology, including GPS, sensors, and data analytics, reshaped farming practices [22].
- Precision farming empowered farmers to precisely targeting fertilizer and pesticide application for enhanced yields.
- Advances in biotechnology have enabled the genetic modification of crops, boosting their productivity and bolstering resistance to diseases.
- Limitation of this stage:
  - ▶ Dependency on inputs: reliance on high-yielding varieties, chemical fertilizers, and pesticides creates economic unsustainability and soil degradation.
  - ▶ High initial investment costs: posing a barrier to entry for small-scale farmers [19].

2.3.4 Agriculture 4.0: Smart Agriculture

Agriculture 4.0 is a transformative concept that integrates precision farming principles with modern information technologies, such as the Internet of Things and Big Data analytics, to enhance decision-making, increase competitiveness, and create a technological value chain that integrates all actors in agri-food production [23].

This signifies the present and future state of agriculture, commencing approximately from 2017 [17] [1]. It focuses on:

- Agriculture 4.0 emerged around the early 2010s, aligning with Industry 4.0's innovative strategy.
- Precision Agriculture evolves into Agriculture 4.0, emphasizing sustainability and efficiency.

- It leverages cutting-edge technologies like IoT, AI, and big data for monitoring and control farming in real-time.
- The integration of sensors, robots, and AI, including machine learning, enables advanced data analysis.
- Agriculture 4.0 aims for global improvements in productivity, food systems, climate adaptation, and environmental impact reduction [24].
- The main challenges associated with Agriculture 4.0 implementation include:
    - Small and medium enterprises operating in agriculture face difficulty in adopting Industry 4.0 innovations due to rapid technological advancement, which makes monitoring and implementation complex.
    - Higher costs for small and medium-sized companies compared to large companies [25].
    - Technological Barriers: the need for infrastructure development in rural areas pose significant obstacles to the adoption of Agriculture 4.0 [23].
- Raises security concerns regarding data privacy and vulnerability to cyberattacks.

It's crucial to tackle these obstacles and hurdles to unlock the full advantages of Agriculture 4.0, paving the way for a sustainable, efficient, and resilient agricultural sector.

2.4 Importance of Smart Agriculture in Modern Farming Practices

Smart Agriculture plays a crucial role in human beings life due to its importance on enhancing his profits to resolve challenges of traditional agriculture.

In this section we introduce some key reasons for the befits of SA:

- Climate Resilience: smart agriculture practices help farmers adapt to and mitigate the impacts of climate change on agricultural production.
- Efficiency: by utilizing advanced technologies like IoT, drones, and data analysis systems, farmers can optimize resource use, leading to increased efficiency in farming operations.
- Precision Farming: smart agriculture enables precision farming techniques, allowing for targeted application of inputs such as water, fertilizers, and pesticides based on real-time data, leading to higher yields and reduced environmental impact.
- Sustainability: smart agriculture promotes sustainable farming practices by enhancing productivity while minimizing resource wastage and environmental degradation.
- Data-Driven Decision Making: by collecting and analyzing data on soil quality, weather conditions, and crop health, farmers can make informed decisions to improve crop management and overall performance.

- Global Impact: smart agriculture practices contribute to reducing greenhouse gas emissions and combating climate change on a global scale.

- Technological Innovation: smart agriculture drives innovation in the agricultural sector, fostering the development of new technologies and solutions to address the challenges faced by farmers.

- Competitiveness: sdopting smart agriculture practices enhances the competitiveness of farmers by improving productivity, reducing costs, and ensuring sustainable agricultural practices.

- Long-Term Sustainability: smart agriculture technologies are seen as a key factor in ensuring the long-term sustainability of agricultural activities, both at the national and international levels.

- Future Prospects: smart farming technologies are projected to play a crucial role in shaping the future of agriculture, with a focus on sustainability, efficiency, and resilience to climate change impacts [26] [23].

2.5 The integration of ICTs in agriculture

ICT is defined by UNESCO as “the combination of informatics technology with other, related technologies, specifically communication technology”.

The integration of ICTs in agriculture involves adopting digital technologies like IoT, AI, and big data analytics to enhance productivity, sustainability, and decision-making processes in farming. This adoption is driven by the potential of smart agriculture to address agricultural challenges, boost crop yields, and reduce costs [27]. By employing artificial intelligence, research and development in smart farming are accelerated. Digital technologies such as wireless sensor networks and cyber-physical systems are integrated into traditional agriculture systems to improve agility, resource efficiency. While digital technologies offer strategic solutions for increasing farm output efficiency, challenges related to technological, socioeconomic, and management aspects need to be addressed for the full realization of agricultural 4.0 [28].

Cyber-physical systems (CPSs) seamlessly blend computing with physical operations, integrating sensors, models, and actions to facilitate closed-loop setups. These systems harness data from the physical world to inform decisions in the digital realm, carefully considering constraints like time, energy, and safety [29]. Leveraging cutting-edge technologies such as sensors, machine learning, and cloud computing, CPSs enable predictive analytics and optimize performance for efficient operation. Interconnected through the Internet, lightweight and portable computing devices empower CPSs to monitor and control systems in real-time, ensuring proper operation and swift responses. By managing thousands of devices in coupled environments, CPSs offer enhanced convenience in control and management tasks, driving advancements in various fields [30] [31].

Cyber-Physical-Social Systems (CPSS) expand CPS to include social interactions, forming the basis for smart applications across various domains [32].

2.6 Overview of Greenhouse Farming

Greenhouses are essential structures in modern agriculture, because they optimize plant growth and ensure consistent yields by regulating factors like temperature and humidity.

2.6.1 Greenhouse definition

Greenhouse farming stands out as an epitome of precision and sustainability within the realm of smart agriculture [33]. A greenhouse is a specialized structure, resembling a house, coated with materials like plastic or glass, facilitating the year-round cultivation of crops like showing in figure 2.2. It serves to manipulate plant growth conditions, ensuring increased yields and improved quality of produce. Traditional greenhouse designs typically neglect environmental factors such as temperature and humidity variations. However, modern greenhouses adopt smart technologies and monitoring systems to regulate these variables effectively. The functionality of a productive greenhouse relies on the implementation of environmental control devices to manage various weather parameters. By integrating these technologies, greenhouses can optimize plant growth and enhance agricultural productivity [11] [34].

Figure 2.2: Greenhouse [2]

2.6.2 Greenhouse farming techniques

The cultivation techniques of greenhouses play a crucial role in the productivity and efficiency of agricultural operations. Understanding the challenges and benefits associated with each method is essential for informed decision-making in greenhouse farming. (see Table2.1)

Greenhouse Farming Method	**Knowledge about It**	**Challenges**	**Benefits**
Less Developed Greenhouse (LDG)	- Permanent greenhouse technique suitable for flowers and vegetables.	- Poor environmental control. - Fewer investment charges. - Lacks ventilation, limiting applicability in hot regions.	- Constructed from readily available wooden materials, making it affordable for agriculturists with limited budgets. - Flexibility in structure expansion.
Average Developed Greenhouse (ADG)	- Constructed with polythene or glass sheets. Ventilation methods can be static or movable. cost ranges from $30 to $100 per square meter.	- Requires initial investment. - Maintenance costs may be moderate.	- Improved well-organized environmental control system. - more flexibility than LDG. - Capable to manageable in hot regions. - Lower power consumption. - Suitable for cultivating vegetables and high quality plants.
Highly Developed Greenhouse (HDG)	-Minimizes labor costs through task automation. - Constructed with a glass/iron structure. -Internal weather is independent of external atmosphere. - Climate controlled by monitoring factors like temperature, CO2 level, and humidity. - suggesting using it for cold-winter areas and nursery production.	- High maintenance costs exceeding $100 per square meter. - Initial investment may be prohibitive for some farmers.	- Labor cost reduction through task automation. - Advanced climate control. - Suitable for cold-winter areas and nursery production. - Efficient for enhancing crop quality and yield rates. - Suitable for cold-winter areas and nursery production. - eco-friendly

Table 2.1: Greenhouse Farming Methods [11]

2.6.3 Insect pests in Greenhouses

In this section, we introduce some basic information about sources of Pests in Greenhouses, the definition of plant diseases and pests, Common pests found in greenhouse environments and the Impact of harmful insects on greenhouse, then we talk about the different control methods for common pests found in greenhouse.

Greenhouses provide an ideal environment for plant growth, but they also attract various harmful insects that can jeopardize crop health.

The insect pests inflict significant damage, with global annual yield losses estimated at 18–20%, valued at over US$470 billion. In Indian agriculture, losses reach 30–35%, costing around US$36 billion, impacting agricultural markets, food security, and farmers' profits. Efforts in pest management, including the use of transgenic crops, have seen a decline in losses, yet challenges persist, especially with climate change impacting pest behavior and crop yields [35].

2.6.3.1 Definition of plant diseases and pests

Plant diseases and pests constitute natural disaster and severe threats to worldwide agriculture and forestry, disrupting the normal growth of plants from seed development to seedling growth, often leading to plant death. These afflictions adversely impact various stages of plant development, posing significant challenges to agricultural productivity and crop health [36] [37]. So Knowing the location, extent and severity of the occurrence of diseases and pests is essential in guiding plant protection procedures [38].

2.6.3.2 Different sources of pests in greenhouses

It's crucial to understand the diverse origins of these intruders. Below, we outline the primary sources:

- Impact of Global Warming: the shifting climate patterns wrought by global warming can prompt alterations in pest distribution, potentially facilitating the survival of exotic species in temperate climates.
- Cross-Contamination Among Greenhouses: in temperate greenhouse environments, exotic pests often proliferate through cross-contamination among neighboring facilities, rather than originating from surrounding vegetation. These pests may exploit outdoor vegetation for propagation during the summer months.
- Aerial Dispersal of Indigenous Pests: indigenous pests infiltrate greenhouses via aerial dispersal mechanisms. Certain species possess the ability to traverse considerable distances on air currents, frequently gaining entry through ventilation openings.
- Human-Mediated Introduction via Workers and Nursery Plants: human activity, particularly the inadvertent transport of insects by greenhouse workers on their attire and belongings, represents a significant vector for pest introduction. Additionally, the introduction of nursery plants and associated materials can serve as a conduit for infestations within greenhouse environments [39].

2.6.3.3 Common pests found in greenhouse environments

In greenhouse environments, harmful insects like aphids, whiteflies, thrips, leaf miners and more pose significant threats to crop health [40] as showing in figure 2.5. They inflict damage by feeding on plant tissues, transmitting diseases, and weakening plant vigor and yield. Employing integrated pest management techniques such as insect nets, biological control agents, and targeted insecticides can effectively mitigate these pests' impact while reducing reliance on chemical pesticides [41].

Unbalanced and excessive application of insecticides will contribute to another big issues, because this indiscriminate use lead to the development of resistance in various pest species, undermining the effectiveness of pest control measures and necessitating alternative management strategies. [42].

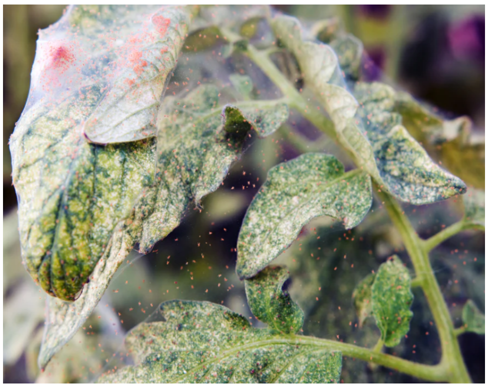


Figure 2.3: The damage of spider mites

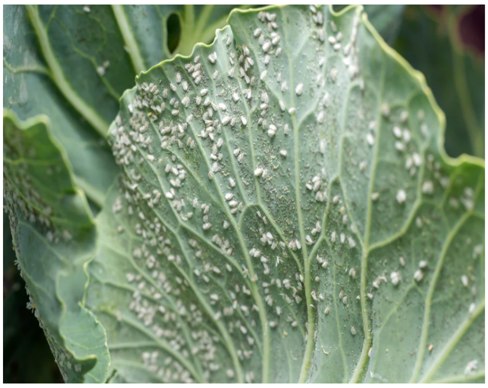


Figure 2.4: The damage of Whitefly

Figure 2.5: Pests damage on crops [2]

2.6.3.4 Impact of harmful insects on greenhouse

The impact of common pests found in greenhouse environments, such as aphids, thrips, mites, whiteflies, slugs, and leafminers, manifests in various detrimental effects:

- Aphids: feed on plant sap, causing stunted growth, distorted leaves, and transmission of plant viruses.(Figure 2.6)

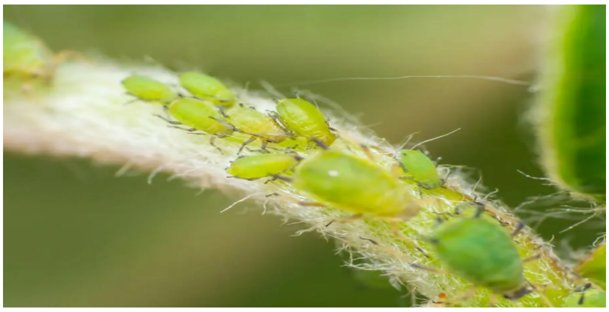


Figure 2.6: Aphids [3]

- Thrips: feed on plant tissues, leading to stippling, silvering, and distortion of leaves, along with disease transmission.

- Whiteflies: sap feeders causing yellowing, wilting, and inhibited growth, also promoting sooty mold growth through honeydew secretion [40]. (Figure 2.7)

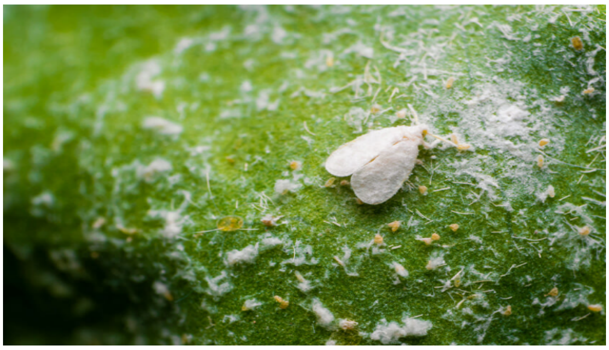


Figure 2.7: Whitefly [3]

- Mites: damage plants by feeding on cells, resulting in stippling, discoloration, and webbing, reducing plant vigor and yield.
- Slugs: cause extensive damage by feeding on leaves, stems, and fruits, leading to holes, slime trails, and reduced marketability.
- Leafminers: tunnel within leaves, creating unsightly mines, interfering with photosynthesis, reducing yields, and making plants susceptible to diseases [12].

In summary, harmful insects pose a serious threat to greenhouse farming by causing direct crop damage, transmitting diseases, disrupting biological control measures, necessitating chemical control interventions, and resulting in substantial economic losses.

2.6.3.5 Control methods for common pests found in greenhouse

In this section, we talk about the various strategies for managing these pests which are: cultural, chemical and biological control.

but before that we will mention Phytosanitation which is crucial in greenhouses to prevent the entry and spread of plant and animal pests. It involves various measures such as installing double access doors, using footbaths with disinfectants, setting up washing stations, removing weeds, disposing of infested plant material properly, and avoiding conditions that promote disease development like water condensation. These practices aim to limit pest establishment and reduce the risk of pest and pathogen movement within and across greenhouse borders [43].

2.6.3.5.1 Cultural and Chemical Control In this section,we mention cultural and chemical methods for common harmful insect pests. (See Table 2.2)

The cultural practices are the most cost-effective options for farmers such as timely land preparation and strategic planting to control insect pests culturally. In addition, intercropping and soil nutrient applications (eg: planting carrots with onions reduce pest attacks such as carrot fly) [40].

For the chemical control, we should think carefully before using the insecticides because it involves diverse deferent types and modes of action, such as induced phytoalexins or constitutive phytoanticipins...etc [44].

Insect pests	**Cultural practices**	**Chemical practices**
Aphids	-Prune infested plant parts and promote plant diversity to attract beneficial insects.	-Use water sprays to physically remove aphids from plants.
Thrips	-Remove weeds and plant debris that can harbor thrips.	-Use insecticidal soaps or oils as a last resort.
Whiteflies	-Use yellow sticky traps to monitor and reduce whitefly populations.	-Apply insecticides if populations exceed economic thresholds.
Slugs	-Use barriers like copper tape to protect plants from slug damage.	-Apply slug baits containing iron phosphate.
Mites	-Increase humidity levels to discourage mite infestations.	-Use miticides selectively to target mite populations.
Leafminers	-Remove and destroy infested plant material to reduce leafminer populations.	-Use insecticides like spinosad or neem oil to manage leafminers

Table 2.2: Cultural and Chemical Control for common insect pests [12]

2.6.3.5.2 Biological Control Its strategies for each pest aim to reduce their populations using natural enemies, providing an environmentally friendly alternative to chemical pesticides in greenhouse agriculture.

- Miridae Bugs: increase predator densities near greenhouses by providing plants like Geranium species or pot marigold, which support Zoophytophagous bugs of the family Miridae [39].
- Aphids:
  1. Parasitoids: aphid parasitoids from the families Aphidiidae and Aphelinidae are important natural enemies used for biological control of aphids in greenhouse crops [45].
  2. Predators: predatory insects that feed on aphids, such as ladybugs and lacewings, are beneficial for controlling aphid populations.
  3. Entomopathogenic fungi: some entomopathogenic fungi have shown potential for controlling aphids in greenhouse settings.
- Whiteflies:
  1. Parasitoids: encarsia formosa and Eretmocerus spp. are parasitoids commonly used for biological control of whiteflies in greenhouse crops.
  2. Entomopathogenic organisms: certain entomopathogenic fungi can also be used to target whiteflies in biological control programs [46].

3. Predators: predatory insects like Macrolophus spp., Nesidiocoris tenuis, and Dicyphus spp. are effective in controlling whitefly populations. (See figure 2.8)

Figure 2.8: Insect predators in greenhouse tomato crops [4]

- Leafminers:

  1. Parasitoids: Omnivorous predators like Zoophytophagus mirid bugs are acknowledged as efficient biological control agents in diverse crops, including tomato [47]. Also, the parasitoid Dacnusa sibirica has been studied for its potential to control leafminer pests in greenhouse crops.
  2. Predators: predatory insects like Gronotoma micromorpha have been explored for biological control of leafminers.

- Spider Mites:

  1. Predators: predatory mites, such as Phytoseiulus persimilis, are commonly used for biological control of spider mites in greenhouse crops [48].
  2. Entomopathogenic fungi: certain entomopathogenic fungi can infect and kill spider mites, contributing to their biological control.

- Thrips:

  1. Predators: predatory insects like Orius spp. and Amblyseius spp. are effective predators of thrips in greenhouse crops.
  2. Entomopathogenic fungi: some entomopathogenic fungi have shown potential for controlling thrips populations in greenhouse settings. [4]

2.7 Conclusion

In conclusion, this chapter provided a comprehensive overview of Smart Agriculture, beginning with fundamental concepts such as the definition of SA and the Evolution of Agriculture. Then, we highlighted the importance of Smart Agriculture in modern farming, emphasizing the crucial role of ICT integration in agricultural systems. Transitioning to Greenhouse Farming, we mentioned its techniques, Focusing on insect pests in greenhouses, where we elucided their origins and the harmful impact they impose on greenhouse crops. In the end, we addressed the common insect pests and various control methods employed to mitigate their effects.

In the next chapter, we will discuss Classification and deep learning Techniques for Insect Detection.

Chapter 3

Classification & DL Technique for Insect detection

3.1 Introduction

The growth in capabilities in AI are expanding the ways that this can be used in fields to automate tasks that historically have been manual (like material site visits and mapping) or to prevent problems altogether from happening This chapter is separated into two parts. In the initial chapter, we would follow the fundamentals of machine learning and deep learning and get deeper insights into it - all through our algorithms and image classification. We will start by explaining the concept of artificial intelligence and then with an introduction to machine learning, explaining its basic definitions and types. In the upcoming ones, we move to the deep gray waters of deep learning, its concepts, its types and the available tools.

The second section covers this last pest of tomato: tomato leaf miner Tuta absoluta, its analog, traits and impact. Next, we will dive into some articles on this insect and do a side by side compare. Finally, we will conclude with a discussion.

3.2 Artificial intelligence (AI)

Artificial intelligence (AI) refers to the field of study and development of computer systems that can perform tasks typically requiring human intelligence. These tasks include speech recognition, decision-making, and pattern identification. AI encompasses various technologies such as machine learning, deep learning, and natural language processing (NLP) [49].

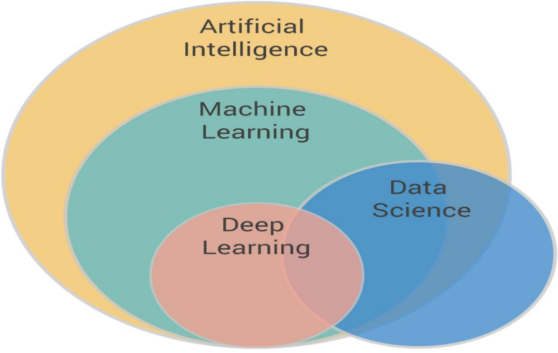


Figure 3.1: AI, ML, Deep learning [5]

3.2.1 Machine learning (ML)

Machine learning, a subset of AI, involves developing algorithms for computers to learn from data without explicit programming, improving performance based on past experiences, and making predictions [50]. These algorithms work on a large dataset containing examples defined by features (nominal, binary, ordinal, or numeric). The data is used to train the algorithms, which then build a model capable of predicting new examples based on the knowledge acquired during training [6].

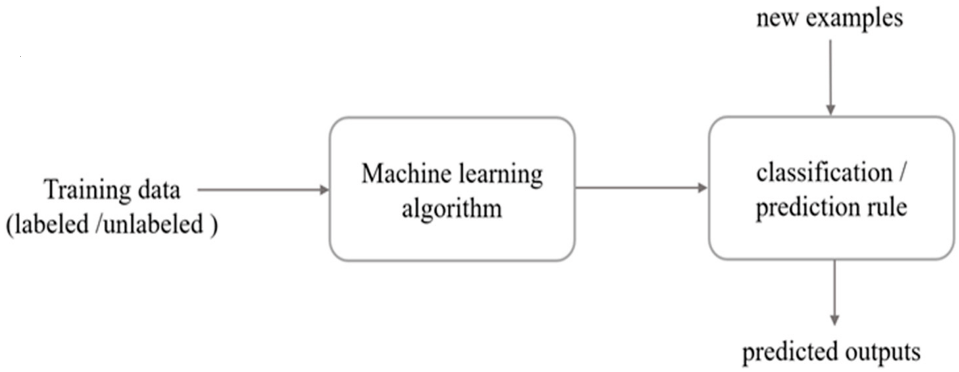

new examples
Training data (labeled /unlabeled)
Machine learning algorithm
classification / prediction rule
predicted outputs

Figure 3.2: A typical machine learning approach [6]

3.2.1.1 Types of Machine Learning Algorithms

There are several divisions of machine learning algorithms, categorized into four groups: Supervised learning, Unsupervised learning, Semi-supervised learning, and Reinforcement learning, as shown in Figure 3.3 .

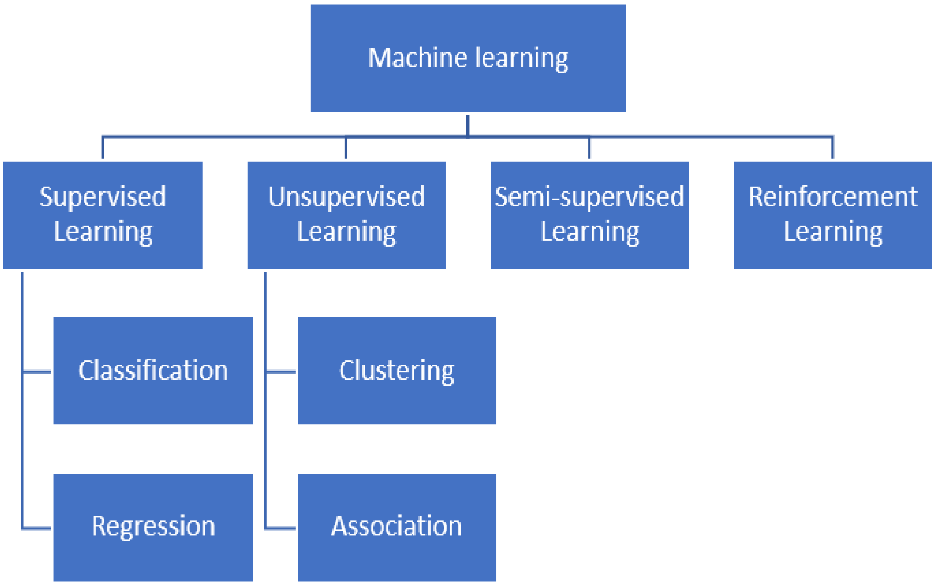


Figure 3.3: Types of Machine Learning Algorithms

3.2.1.1.1 Supervised Learning Supervised learning relies on labeled datasets with input-output pairs to teach machine algorithms, continually supplying data until the model accurately fits it. This enables the model to forecast outputs for new, unlabeled data, making it widely used in domains like medical diagnosis, spam detection, and image recognition for learning patterns and making predictions [51].

- Classification: Identifies categories or classes for input data. For example: determining whether an email is spam or not spam based on its content
- Regression: The algorithm predicts a continuous value. For example, predicting house prices based on factors like area, number of bedrooms, and location.

3.2.1.1.2 Unsupervised Learning Unsupervised Learning: Involves discovering hidden patterns or structures in unlabeled data without specific guidance or predefined outcomes. It is used when labeled data might be scarce or expensive to obtain [52].

- Clustering: Grouping similar data points together based on certain features or characteristics. For example, clustering customers based on their purchasing behavior.
- Association: Discovers relationships and associations between variables or items in a dataset.

3.2.1.1.3 Semi-supervised Learning Semi-supervised Learning: Operates on both labeled and unlabeled data, bridging supervised and unsupervised methods to improve predictions. In many real-world scenarios, obtaining labeled data can be costly or time-consuming, while unlabeled data might be more readily available. This is where semi-supervised learning comes into play [53].

3.2.1.1.4 Reinforcement Learning(RL) It is a type of machine learning where an agent learns to make decisions by interacting with an environment. The agent aims to maximize cumulative rewards through a trial-and-error process. [51].

- Agent: The entity making decisions and taking actions within an environment.
- Environment: The external system with which the agent interacts. It could be a game, a physical system, a simulation, or any other setup.
- Actions: Choices or decisions made by the agent that affect the state of the environment.
- State: The current situation or configuration of the environment.
- Rewards: Feedback provided by the environment to the agent based on its actions. Rewards indicate how good or bad the actions were in a particular state.

3.2.2 Deep learning (DL)

Deep learning (DL) is a subset of machine learning (as shown in figure 3.1) that involves the use of artificial neural networks (ANNs) with multiple layers (hence ”deep”) to learn and understand complex patterns in data. It is designed to mimic the way the human brain processes and learns from information, allowing machines to recognize patterns, classify data, and make predictions or decisions [54].

3.2.2.1 Types of Deep Learning

Deep learning (DL) encompasses various approaches to training neural networks to learn from data. These approaches can be broadly categorized into supervised learning and unsupervised learning, each serving different purposes and utilizing different types of data.

3.2.2.1.1 Supervised Learning: Supervised learning is a task-driven approach that uses labeled training data to train models to make predictions or classifications based on input data. In DL, supervised learning involves deep networks designed for supervised or discriminative learning. These networks typically consist of multiple layers that process information hierarchically to learn and extract patterns from the data [55].

3.2.2.1.2 Unsupervised Learning: Unsupervised learning is a machine learning technique where the neural network learns to discover patterns in data without labeled target variables. DL algorithms like autoencoders and generative models are used for unsupervised tasks such as clustering, dimensionality reduction, and anomaly detection. Unsupervised learning falls under the category of deep networks used for unsupervised or generative learning in DL techniques [56].

Both types of DL utilize neural networks with multiple layers to learn complex patterns from data. They differ primarily in the kind of data they use during training [57].

3.2.2.2 Types of Data

3.2.2.2.1 Structured Data: Consists of tables or spreadsheets containing numerical or categorical data arranged in rows and columns. Deep learning models can learn efficiently on tabular data and allow us to build data-driven intelligent systems. These models can handle large datasets and complex relationships within the data, making them valuable for tasks such as regression, classification, and recommendation systems [58].

3.2.2.2.2 Unstructured Data: Includes text, images, audio, and other formats that lack a fixed structure. Deep learning models can process unstructured data after being trained and reaching an acceptable level of accuracy.

3.2.2.2.3 Semi-Structured Data: Combination of structured and unstructured data, such as JSON or XML files. Deep learning models can process semi-structured data after being trained and adapted to the specific structure [58].

Benefits of working with different data types in deep learning include automatic feature learning, improved accuracy, and reduced reliance on human intervention. Additionally, deep learning models can benefit from data preprocessing techniques to ensure data quality and minimize human effort.

3.2.2.3 Techniques in Deep Learning

Below are several types of deep learning techniques that can effectively and reliably solve issues that are too difficult for the human brain to solve.

1. **Classic Neural Networks:**

   These networks, also known as fully connected neural networks, are the simplest type. They consist of input, hidden, and output layers, with each neuron in one layer connected to every neuron in the next layer.

2. **Convolutional Neural Networks (CNNs):**

   CNNs are highly effective for image processing tasks because they can automatically learn spatial hierarchies of features. They consist of convolutional layers, pooling layers, and fully connected layers.

3. **Recurrent Neural Networks (RNNs):**

   RNNs excel in sequential data processing tasks like natural language processing and time series analysis. Their unique feedback loop allows information to persist over time.

4. **Generative Adversarial Networks (GANs):** It consists of two neural networks that compete against each other. One network generates fake data, while the other tries to distinguish between real and fake data.

5. **Transfer Learning:**

   This technique involves using a pre-trained model as a starting point for a new task. It can save time and resources by leveraging knowledge from previous tasks.

6. **Learning Rate Decay:**

   Learning rate decay is used to adjust the learning rate during training to improve model performance. It involves gradually reducing the learning rate over time [59].

7. **Dropout:**

   Dropout is a regularization technique that helps prevent overfitting in neural networks. It works by randomly dropping out some neurons during training, forcing the network to learn more robust features [59].

3.3 Classification Algorithms

3.3.1 Definition of Classification

Classification is a supervised machine learning method where the model tries to predict the correct label of given input data. In classification, the model is fully trained using the training data, and then it is evaluated on test data before being used to perform prediction on new unseen data [60].

3.3.2 Types of Classification Algorithms for Machine Learning

1. **Logistic regression** Logistic regression, akin to linear regression, is applied when the dependent variable represents categories like ”yes/no.” Despite its name, it functions for classification by categorizing the dependent variable into specific classes based on regression analysis.

2. **K-Nearest Neighbors (KNN):** The K-NN algorithm, a straightforward classification method, categorizes new data points by identifying their proximity to existing classes. Operating as a non-parametric, lazy learning approach, it relies on similarity measures, such as distance functions, for classification.

3. **Support Vector Machines (SVM):** SVMs, a supervised machine learning model, excel in classification and regression by determining a hyperplane, the decision boundary, with the maximum margin. Versatile, they adeptly handle both linear and non-linear input spaces, making them valuable for high-dimensional data.

4. **Naive Bayes** is a probabilistic classifier that assumes the presence of a certain feature is independent of other features. It is simple, easy to implement, and performs well in many applications, such as spam detection and sentiment analysis.

5. **Decision trees** Decision trees construct tree-shaped models for classification or regression by iteratively splitting datasets based on the Iterative Dichotomiser 3 (ID3) algorithm, resulting in decision nodes and leaf nodes [60].

3.4 Image classification

3.4.1 Definition of image Classification

Image classification is the task of categorizing and assigning labels to groups of pixels or vectors within an image dependent on particular rules. The categorization law can be applied through one or multiple spectral or textural characterizations [61].

Image classification techniques are mainly divided into two categories: Supervised and unsupervised image classification techniques.

3.4.1.1 Unsupervised classification

Unsupervised classification, a fully automated technique devoid of training data, employs machine learning algorithms to analyze and cluster unlabeled datasets. This method discerns hidden patterns or data groups without human intervention, systematically recognizing image characterizations during processing [62].

Two popular algorithms used for unsupervised image classification are ‘K-mean’ and ‘ISODATA.’

- **K-MEAN** clustering objects into k groups based on characteristics.
- **ISODATA** using iterative self-organizing data analysis techniques, allowing flexibility in the number of clusters [63].

3.4.1.2 Supervised classification

Supervised image classification relies on labeled reference samples to train the classifier for categorizing new data. Users visually select training samples within the image, assigning them to predefined categories like vegetation, roads, water, and buildings. This process establishes statistical measures applied across the entire image [62].

3.4.2 Image Classification Techniques

Image classification relies on various methods to categorize images based on their features. Two common approaches include ”maximum likelihood” and ”minimum distance” classification. In ”maximum likelihood” classification, statistical properties like the standard deviation and mean values of textural and spectral indices are initially analyzed. The likelihood of each pixel belonging to specific classes is then calculated using a normal distribution for pixels within each class. This process integrates classical statistics and probabilistic relationships to assign pixels to the class with the highest likelihood [61].

3.4.3 The Process of Image Classification

From a computer’s perspective, an image is interpreted as an array of matrices, with pixels related to the image’s resolution. Image classification involves algorithmic analysis of this statistical data. In digital image processing, pixels are grouped into predefined categories, referred to as ”classes.” Algorithms identify and segregate prominent features, reducing the workload on the final classifier. This feature extraction process significantly influences subsequent categorization steps. Supervised image classification heavily relies on the input data's quality; a well-optimized dataset with balanced classes and high-quality images performs better than a poorly curated dataset with imbalances and subpar annotations [61].

3.4.4 Objective in Image Classification

Transformers excel in image classification by efficiently handling variable-sized inputs and capturing global dependencies within images. Key aspects of their application in image classification include:

- **Eliminating the need for cropping or resizing:** Transformers eliminate the need to crop or resize images with varying resolutions, as they effectively handle inputs of variable sizes.
- **Modeling global dependencies:** Transformers can recognize patterns that extend across the entire image, capturing global context effectively.
- **Adaptability:** Vision Transformers (ViTs) demonstrate remarkable adaptability, showing promise in various computer vision tasks, particularly image classification.
- **Pre-training:** ViTs achieve state-of-the-art performance through pre-training on extensive datasets, followed by fine-tuning on task-specific datasets.
- **Data efficiency:** Data-efficient Image Transformers (DeIT) enhance the efficiency of deep learning models by using data more judiciously [64].

3.4.5 Machine Learning in Image Classification

Machine learning for image recognition involves using algorithms to extract latent insights from structured and unstructured datasets, primarily through supervised learning methods.

Deep learning is a prominent technique in machine learning, characterized by the incorporation of numerous hidden layers within a model, enabling the automatic discovery and representation of complex patterns and features. This advanced approach has proven highly effective in tasks requiring a deep hierarchical understanding of visual data [61].

3.4.6 Object Detection

Object detection, a computer vision challenge, entails the identification and localization of objects in an image using bounding boxes. While traditionally associated with natural language processing (NLP), transformers, a category of deep learning models, have demonstrated adaptability for object detection tasks as well. Their application in this context showcases the versatility of transformer models beyond their conventional use in NLP. This adaptation highlights the evolving landscape of deep learning techniques in diverse domains, extending their impact beyond traditional applications. [65].

3.4.6.1 Transformers Application in Object Detection

In Vision Transformer (ViT), images are separated into patches, treating each patch as a token. These tokens are then processed by a transformer model, which works on them sequentially. The transformer's output consists of embeddings, with each patch represented by a corresponding embedding.

To adjust ViT for object detection, additional learnable parameters are incorporated into the model to predict object bounding boxes and labels within the image. This is achieved by introducing a multi-head self-attention mechanism, enabling the model to focus on different image regions when determining object positions.

Another approach in using transformers for object detection is exemplified by the DETR (DEtection TRansformer) model. DETR utilizes a transformer-based architecture to directly predict object detections without relying on anchor boxes or region proposal networks [66].

3.4.6.2 Object Detection Focus

- Transformers have shown promise in object detection applications.
- Their superiority over traditional models stems from their adaptability to varying input sizes.
- Transformers' self-attention mechanisms improve the ability to capture intricate spatial relationships among objects.
- This capability is particularly beneficial in image analysis.

3.4.7 Image Segmentation

Image segmentation is a computer vision task that involves partitioning an image into multiple segments or regions, each representing a distinct object or image part. While transformers are predominantly known for their success in natural language processing tasks, they have also been explored for image segmentation applications [67].

3.4.7.1 Image Segmentation Objective

- Transformers excel in image segmentation tasks due to their adaptability to varying input sizes.
- Another advantage is their ability to capture global context information, enhancing overall segmentation accuracy.
- However, a challenge arises in balancing the trade-off between local and global information, critical for precise segmentation.
- Additionally, the computational burden of processing large images poses a significant obstacle to the widespread adoption of transformers for this purpose.

3.5 Case Study: Tuta absoluta

Tuta absoluta is a devastating pest that poses a significant threat to tomato crops worldwide. In our work, we will focus on the concept and characteristics of Tuta absoluta, also known as the tomato leaf miner, as well as its effects on tomato plants. We will explore the statistics regarding the spread of Tuta absoluta and investigate various solutions aimed at eliminating this pest to protect tomato crops.

3.5.1 Definition of Tuta absoluta

The Tuta absoluta, commonly known as the tomato leafminer, is a destructive moth species that poses a significant threat to tomato crops. The larvae of this pest feed voraciously on tomato plants, damaging leaves, stems, and fruits. Originating from South America, Tuta absoluta has spread globally, causing substantial economic losses in agriculture. Effective pest management strategies are essential to mitigate the impact of this invasive species on tomato cultivation [68].

The adult tomato leafminer (Tuta absoluta) is a small butterfly native to South America and comes to Algeria in 2008 [69]. Its larvae enter the leaves, fruits, and branches in order to feed.

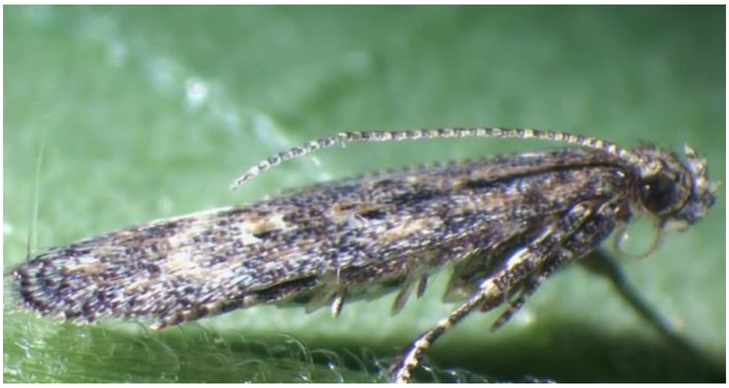


Figure 3.4: Tuta absoluta [7]

3.5.2 The phenology of Tuta absoluta

The phenology of Tuta absoluta,also known as the tomato leafminer, refers to its life cycle stages and the associated events.

Tuta absoluta has a life cycle that consists of four main stages: egg, larvae, pupa, and adult.

3.5.2.1 Egg Stage

Eggs of Tuta absoluta are small, cylindrical, and creamy-white to yellow, measuring approximately 0.35 mm in length. They are laid by adult females on host plants, mainly on the underside of leaves, stems, and petioles [70].

The duration of egg development varies with temperature, ranging from 4.0 to 11 days at different temperatures [71].

A mature female can lay up to 260 eggs during her life cycle. Egg hatching occurs 4-6 days after being laid.

3.5.2.2 Larvae Stage

The larvae of Tuta absoluta are cream-colored with a characteristic dark head. They are the most damaging stage to crops as they feed on plant tissues by creating mines, leading to its damage, leaf drying, and defoliation.

Larval development consists of four instars, with pupation potentially taking place in the soil, on leaf surfaces, or within mines [70].

The larval stage duration ranges from 6.3 to 16.0 days at different temperatures [71].

3.5.2.3 Pupal Stage

Pupation in Tuta absoluta may occur in the soil, on leaf surfaces, within mines, or in packaging material. A cocoon is built if pupation does not occur in the soil.

The total life cycle of Tuta absoluta is completed within 30-35 days under suitable conditions [70].

3.5.2.4 Adult Stage

Adult Tuta absoluta are small moths with a body length of 5-7 mm. They have silvery-brown coloration and can be identified by their thread-like antennae and forewings with grey scales and black spots.

Adults are nocturnal and hide between leaves during the daytime. They reproduce rapidly, with a life cycle ranging from 24 to 38 days.

The moths are active during the night, and adult females lay eggs on host plants to initiate a new generation [70].

Females mate once a day and can mate up to six times during their lifespan, with a single mating bout lasting 4-5 hours.

Females lay eggs primarily 7 days after the first mating, with a maximum lifetime fecundity of 260 eggs per female [71].

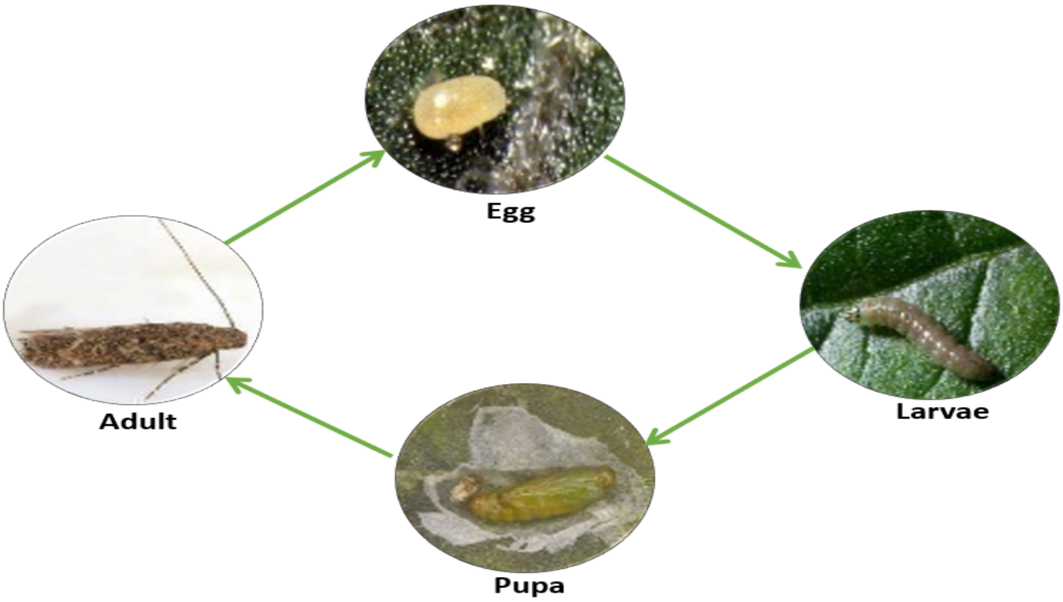


Figure 3.5: Life cycle of T. absoluta [8]

3.5.2.5 Population Dynamics

Tuta absoluta is a multivoltine species with a high growth population rate.

The pest can complete multiple generations per year in tropical regions with temperatures between 25 and 30 °C.

The intrinsic rate of natural increase, gross reproductive rate, and net reproductive rate are significantly higher at temperatures between 20-25 °C [72].

3.5.3 Characteristics of tuta Absoluta

- It is a difficult pest to control due to its ability to hide under leaves and develop resistance to pesticides.
- Tuta absoluta is oligophagous and can survive and reproduce normally on potatoes, tobacco, and other Solanaceae crops.
- It has a strong preference for tomatoes among host plants, and volatile chemical signals play important roles in its host plant preferences.
- The larvae of Tuta absoluta attack leaves, buds, stems, flowers, calyces, and tomato fruit, causing crop losses up to 80-100% in the absence of timely control measures.
- Tuta absoluta has a high reproductive capacity and can develop multiple generations per year.
- The use of insecticides has limited effectiveness due to the pest's endophytic feeding behavior, making Tuta absoluta a difficult target for insecticide sprays [73].

3.5.4 The effect of Tuta absoluta on tomatoes

- Tuta absoluta, also called the South American tomato leafminer, presents a serious threat to tomato cultivation, leading to substantial losses of 80% to 100% when not effectively controlled.
- Damage to Tomato Plants: As a leafminer, Tuta absoluta inflicts damage on multiple parts of the tomato plant, including leaves, terminal buds, flowers, and fruits. The larvae tunnel into and consume the leaves, while also burrowing into the fruits, resulting in significant economic losses. [74].

3.5.5 Statistics of the spread of Tuta absoluta

Tuta absoluta, also known as the South American tomato leafminer, was first identified in Peru by Meyrick in 1917 and has been recorded in all South American countries. It was first introduced into Europe in Spain in 2006, from where it is believed to have spread to other European countries and its range has expanded.

The pest has been recorded in more than 90 countries and regions outside South America, including several African countries such as Algeria, Angola, Botswana, Burkina Faso, Egypt, Ethiopia, Kenya, Libya, Malawi, and Morocco,Also the continent of Europe and Asia [75].

The first outbreaks of Tuta absoluta in Algeria were identified during spring 2008 in the protected tomato crops of Mostaganem region. Subsequently, the infestation extended to beet, spinach, and various weeds in Biskra [76]. (See 3.6)

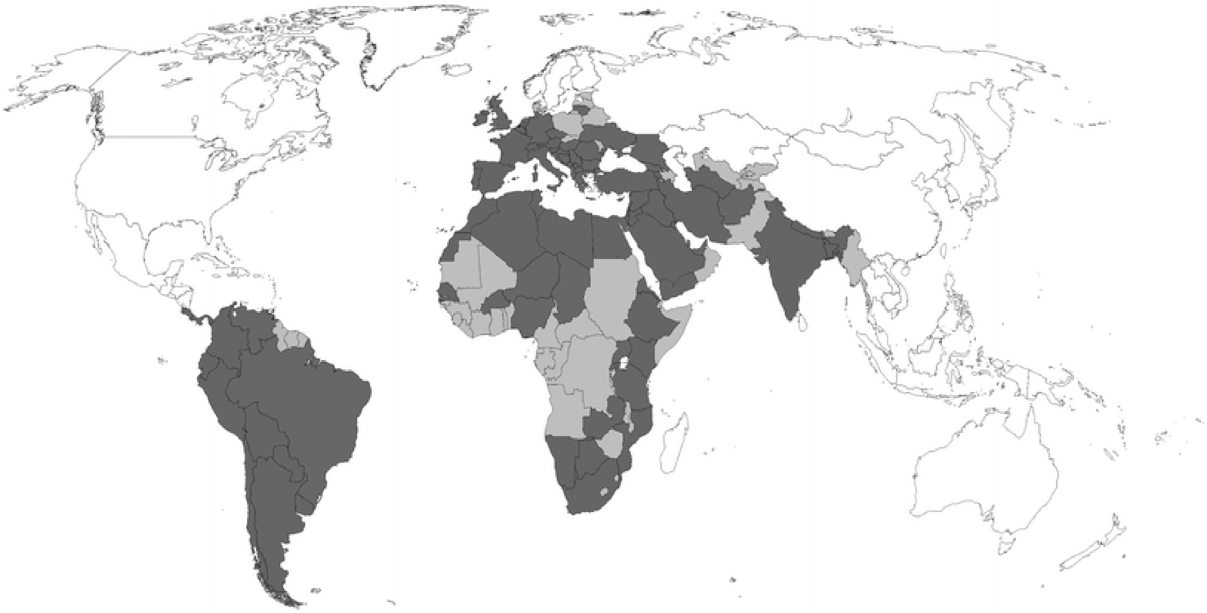


Figure 3.6: Global geographical distribution of the spread of Tuta absoluta [9]

Current global distribution map of Tuta absoluta (3.6 as of April 2017): Countries are marked in two categories: (1) Confirmed presence of T. absoluta (dark gray), and (2) Potential presence of T. absoluta due to geographical and ecological proximity or unconfirmed presence after an initial report (light gray) [9].

3.5.6 Solutions to eliminate Tuta absoluta

- Managing Tuta absoluta presents challenges due to its adeptness at concealing itself beneath leaves and developing resistance to pesticides. which requires only two applications of each pesticide. The most efficient and sustainable approach to tackle this pest involves integrating ecologically acceptable methods, encompassing cultural, biological, and chemical control strategies.

- While chemical control methods can be employed to address Tuta absoluta infestations, their usage should be judicious and complemented by other control measures to mitigate the risk of resistance development. Notable chemical options include insecticides like Tracer and NeemAzal.

- Utilizing invertebrate biological control agents and biopesticides offers a natural and effective means of managing Tuta absoluta. Parasitic wasps such as Trichogramma and predators like Macrolophus can be deployed to regulate the pest population.

- Timely identification of Tuta absoluta is paramount for efficient management. Implementing phytosanitary measures, such as greenhouse disinfection, and closely monitoring the population dynamics of biological control agents, proves instrumental in prevention and control efforts [77].

3.6 Related work

In our comprehensive investigation, we thoroughly explored three scientific articles focused on the Tuta absoluta insect, each contributing valuable insights into its characteristics and potential control measures. These articles are as follows:

1. **G. D. Arturo Cocco et al [78]:**This study evaluated the effectiveness of mass trapping using light and pheromone traps to manage Tuta absoluta infestations in greenhouse tomato crops. Light traps were more effective in reducing leaf damage at low/moderate pest densities during the summer-winter season. Pheromone traps showed varying results in reducing leaf damage and fruit infestation across different trials and densities. The study highlights the potential of mass trapping as a non-insecticide control strategy for Tuta absoluta, with light traps showing promising results under specific conditions.

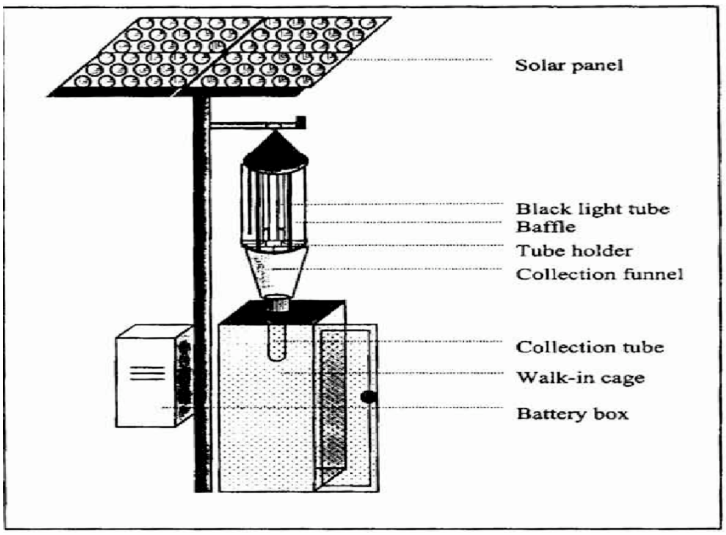


Figure 3.7: Alternate Energy Light Trap [10]

2. **R. M.Raphael Njurai Miano et al [79] :** The study investigates the chemical signals emitted by wild tomato plants that repel Tuta absoluta females, also known as tomato leafminer moths. The researchers use a combination of electroantennogram (EAG) and machine learning techniques to identify the specific volatile compounds responsible for triggering avoidance behavior in the moths.

   The EAG technique measures the electrical activity of the moths' antennae in response to different odor stimuli, helping to identify which compounds elicit a response from the insects' olfactory receptors. Machine learning algorithms are then applied to analyze the EAG data and identify the volatile blend that repels the moths.

   The results of the study likely include the identification of specific volatile compounds or a blend of compounds that mediate avoidance behavior in Tuta absoluta females. This research contributes to understanding the chemical ecology of plant-insect interactions and may have implications for developing natural pest control methods based on the identified volatile blend.

3. **D. M. L. Loyani et al [80] :**The study presents a novel approach using deep learning for segmenting and identifying the damage caused by Tuta absoluta. The researchers developed a mobile application that allows users to capture images of tomato leaves and then uses a deep learning model to automatically detect and segment areas of damage caused by the pest.

   The deep learning model is trained on a dataset of images annotated with the location and extent of Tuta absoluta damage. The model learns to recognize patterns and features associated with the damage, enabling it to accurately identify and segment affected areas in new images. The mobile application provides a user-friendly interface for farmers or researchers to quickly assess the level of damage to tomato plants in the field. By automating the process of damage detection, the application can help improve pest management strategies and ultimately reduce the impact of Tuta absoluta on tomato crops.

In the following sections, we present a detailed exploration of each scientific article on tuta absoluta, analyzing aspects such as their focus, methods used, and key findings. Below is a table that synthesizes the comparative insights derived from these articles:

Article Title	**Focus**	**Methods Used**	**Results Obtained**
1. ”G. D. Arturo Cocco et al [78]”	This research investigates how the effectiveness of light and pheromone traps, at various densities, can reduce Tuta absoluta populations and minimize leaf and fruit damage in greenhouse tomato crops.	In Sardinia, mass trapping trials in plastic greenhouses used pheromone and suction light traps. They counted trapped adults and inspected larval mines and fruits weekly to assess effectiveness.	1. Light traps outperformed pheromone traps in capturing female Tuta absoluta and reducing leaf damage at low to moderate population densities. 2. pheromone traps had variable success in reducing damage and infestation.
2. ”R. M.Raphael Njurai Miano et al [79]”.	This study aims to understand the volatile blend that mediates avoidance behavior by Tuta absoluta females (tomato leafminer moths) towards a wild tomato plant. This likely involves identifying specific chemicals or compounds in the plant’s volatile emissions that repel the moths.	1. Electroantennogram (EAG) measures the electrical activity of moth antennae in response to odors, indicating which compounds stimulate their olfactory receptors. 2. Machine learning is probably employed to analyze EAG data, classify responses, or recognize patterns, aiming to identify the volatile blend that deters Tuta absoluta females.	- Identification of specific volatile compounds or blend of compounds that trigger avoidance behavior in Tuta absoluta females. - Possibly, the development of a predictive model using machine learning to understand and predict moth behavior based on the volatile blend emitted by wild tomato plants.
5. ”D. M. L. Loyani et al [80]”	Developing a mobile application based on deep learning for segmenting Tuta Absoluta’s damage on tomato plants.	collecting images of tomato plants with Tuta Absoluta damage, preprocessing these images, and designing a deep learning model for image segmentation.	- High accuracy in segmenting tuta absoluta’s damage on tomato plants. - Development of a functional mobile application for farmers.

Table 3.1: Comparison of Scientific Articles on tuta absoluta

3.7 Discussion

Scientific research continues to advance significantly. However, despite this progress, a definitive solution to eliminate the tomato leaf miner remains elusive.

Our presentation of research findings, such as ”Potential of mass trapping for Tuta absoluta management in greenhouse tomato crops using light and pheromone traps,” sheds light on this issue.

This study discovered that mass trapping with light traps effectively reduced leaf damage at low to moderate pest densities during the summer-winter tomato cycle. Pheromone traps also exhibited some efficacy in reducing leaf damage and fruit infestation in certain trials. However, light traps were ineffective in controlling T. absoluta infestations during the winter-summer season. Additionally, pheromone traps yielded inconsistent results in reducing leaf damage and fruit infestation.

The seconde article, ”Electroantennogram and machine learning reveal a volatile blend mediating avoidance behavior by Tuta absoluta females to a wild tomato plant,” offers valuable insights into plant-insect interactions and chemical ecology. It highlights the potential for developing natural pest control methods based on the identified volatile blend. However, the study also identifies several challenges and limitations:

- Complexity of volatile blends and their interactions: Understanding the full complexity of volatile blends and how different compounds interact with each other may require further research. This complexity could impact the efficacy and reliability of any pest control methods developed based on these blends.

- Practical application of the findings: While the study provides promising results, the practical application of the identified volatile blend for pest control may require additional validation and field testing. Implementing these findings in real-world settings may pose challenges that need to be addressed through further research and development.

The last article studied is ”A Deep Learning-based Mobile Application for Segmenting Tuta Absoluta’s Damage on Tomato Plants.” It offers several advantages:

- Improved accuracy and efficiency in detecting Tuta Absoluta damage: The use of deep learning in a mobile application enhances the accuracy and efficiency of detecting Tuta Absoluta damage on tomato plants, which can lead to better pest management practices.

- Accessibility and ease of use for farmers through a mobile application: By providing a mobile application, the study makes the detection of Tuta Absoluta damage more accessible and user-friendly for farmers, potentially improving early detection and pest control efforts.

However, the study also has its limitations:

- Dependency on the quality and quantity of the dataset: Like other deep learning applications, the effectiveness of the mobile application is dependent on the quality and quantity of the dataset used for training. Insufficient or poor-quality data can lead to less accurate results.

- Computational resources required for training and inference: Training a deep learning model and running inference on a mobile device require significant computational resources. This could be a limitation for users with limited access to such resources.

After thoroughly reviewing these articles, we've engineered a robust solution for detecting and eradicating tomato leafhoppers with precision and efficacy. Our approach integrates cutting-edge technology, combining a motion sensor and a camera strategically positioned at the greenhouse entrance to swiftly detect any presence of these pests.

The motion sensor acts as the first line of defense, instantly detecting any movement within its range, while the camera is automatically activated to send this real-time data to a deep learning model, which swiftly validates the insect's identity.

Upon identification, the system seamlessly executes a command to capture and eliminate the intruder. To further bolster its efficiency, we've strategically placed this device adjacent to a trap, to keep any Tuta absoluta insects near to greenhouse entrance.

In the event of a successful capture, the system promptly notifies the farmer through a dedicated mobile application. This intuitive app serves as a comprehensive tool, furnishing detailed insights into the detected insect and offering real-time camera footage of the tomato crop's status.

By seamlessly integrating advanced technology with proactive pest management strategies, our solution empowers farmers with actionable insights and enables swift intervention, ensuring the health and vitality of their tomato crops.

3.8 Conclusion

In this chapter, we discussed the core concepts of Artificial Intelligence, including Machine Learning and its types, as well as Deep Learning, its types, and techniques. We then delved into machine learning classification algorithms, along with an overview of image classification.

Furthermore, we shifted our focus to the tomato leaf miner, ”Tuta absoluta” . We provided information about this insect, followed by a study of several scientific articles discussing it. We compared five articles in this regard. In the next chapter, we will analyze and design our system (application).

Chapter 4

Design and Contribution

4.1 Introduction

As we mentioned in the last chapter, no ideal solution has been found to completely eradicate the tomato leaf miner. That is why we consider this part to be the most important, due to its focus on the studies that we conducted.

In this chapter, we delve into the design and contribution aspects of our research, focusing on the development of a comprehensive system for detecting Tuta Absoluta . We begin by outlining the methodology employed to construct the proposed architecture, ensuring that each component is meticulously detailed to provide a clear understanding of the system's functionality. The architecture is divided into distinct layers-Physical, Cloud, Treatment, and End-User each playing a crucial role in the overall system. Following the architectural description, we introduce the Tuta Absoluta detection system, highlighting the steps involved in creating prediction (using ML algorithms) and recognition (using CNN) models and the functional and non-functional requirements. Additionally, we present a system analysis model through use case and sequence diagrams, identifying key actors and scenarios to illustrate the system's operation.

4.2 Methodology

We aim to develop an effective real-time pest control system in greenhouse environments. specifically targeting the Tuta Absoluta insect, which is a primary threat to tomato plants and poses a major problem in many countries.

Our solution will revolutionize pest management in greenhouse environments, and by leveraging advanced technologies such as artificial intelligence and IoT, the architecture offers a comprehensive system for real-time monitoring and intervention through early insect detection and automated elimination.

In order to address the challenge of detecting and eliminating Tuta absoluta in greenhouse environments, our methodology involves the following key components and steps:

A. Using deep learning to develop CNN models specifically trained to identify Tuta absoluta in greenhouses.

B. Using Machine learning to develop regression model specifically trained to predict the existence tuta absoluta in greenhouse based on temperature and humidity dataset.

C. A device equipped with some equipment to collect data in real time and eliminate it which are: movement sensor to detect any movement, camera to get data of the insect(dividing the video into sequence of images) so the cnn model can detect, temperature and humidity sensor(dht11) so the classification regression model can predict and insect aspirator to eliminate this insect.

D. develop a mobile application to notify the farmer with each event(detection, prediction and elimination).

4.3 Proposed architecture

In this section, we start by defining the general system architecture then give description for each layer of this architecture.

To achieve our goal, we proposed an architecture for the system breaks down the system into four distinct, well-defined layers, each with specific functionalities. Starting with the physical layer, which includes the camera, DHT11 sensor, ultrasonic sensor, and insect aspirator, this layer is responsible for data acquisition. The collected data is then transmitted to the second layer, the cloud layer, which handles real-time storage and data transmission. The third layer, the processing layer, performs data analysis, insect recognition model, and prediction model. Finally, the end-user layer. The system architecture is illustrated in the figure below 4.1:

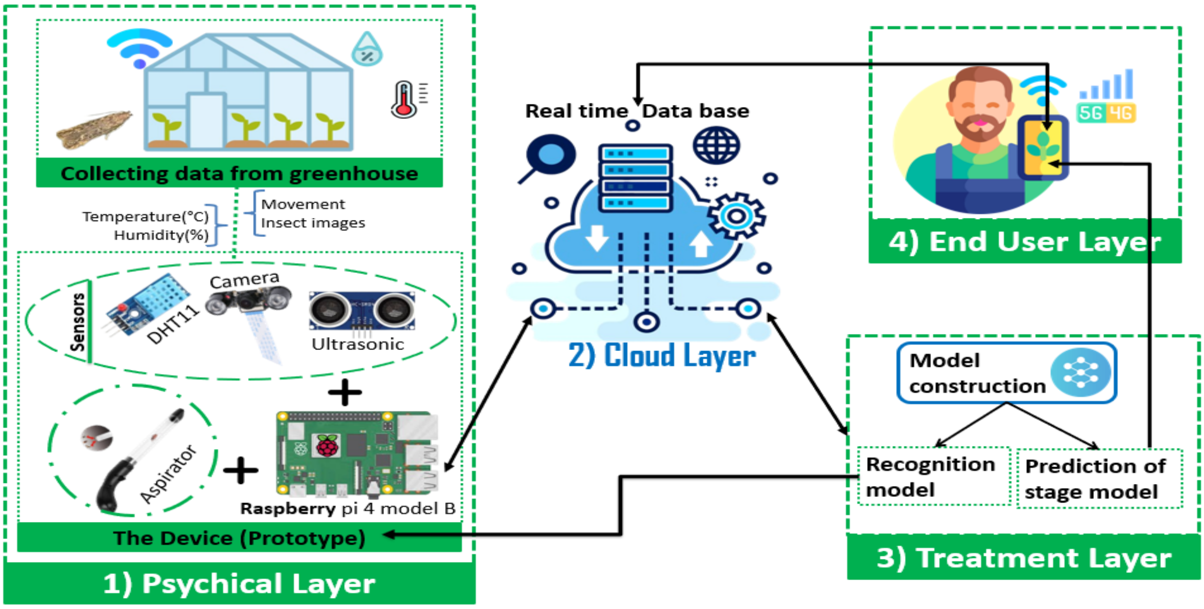


Figure 4.1: The proposed architecture for the system

This approach focuses on real-time analytics. It provides immediate information based on the real-time collected data from the greenhouse. The roles and processes of each layer will be detailed in the next section.

4.3.1 Architecture description

Here is a detailed explanation of each layer in the architecture:

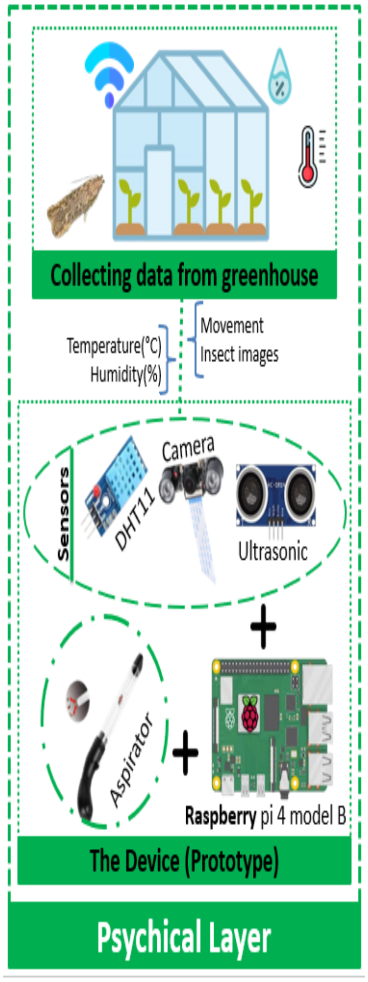
4.3.1.1 Physical Layer:

It is the foundational layer of the system architecture, consisting of the greenhouse environment and our device installed in this greenhouse. This layer is responsible for real-time data collection and initial pest management actions.

We have positioned our device near the greenhouse entrance and the pheromone and light trap (pheromones to lure male tuta absoluta moths. and the light to attract all tuta absoluta moths at night, male and female). this device consists of:

A. Sensors: to collect data from the greenhouse environment in real time, which include:

- DHT11 Sensor: measures temperature and humidity within the greenhouse, providing essential environmental data for pest prediction model(predicting the presence of each stage for this insect).
- Movement Sensor: detects movement within the greenhouse, serving as the initial trigger for the system to activate other components when potential pest activity is detected.
- Camera: get real time data of tuta absoluta from the greenhouse environment, particularly focusing on areas where movement is detected. The video is divided into images, which are used for deep learning-based insect recognition.

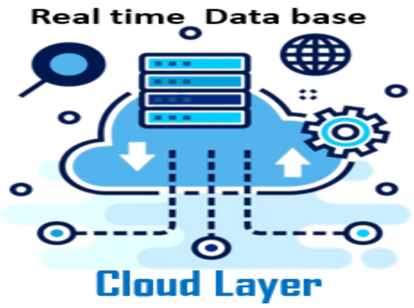


B. Insect Aspirator: activates automatically upon pest detection to capture and eliminate the identified insect, specifically Tuta Absoluta.

C. Raspberry Pi 4 Model B: serves as the central processing unit in the Physical Layer, interfacing with all sensors and the Aspirator. It processes the collected data and coordinates the actions of the other devices.

4.3.1.2 Cloud Layer:

It acts as the intermediary between the physical devices and the data processing and user interaction. It provides the necessary infrastructure for data storage, real-time data processing, and model deployment.



- Real-Time Database: stores data collected from the greenhouse in real-time, including environmental conditions (temperature, humidity) and the images.
- Data Transmission: handles the bi-directional flow of data, sending collected data from

the physical devices to the cloud for processing and receiving commands or updates from the cloud back to the devices.

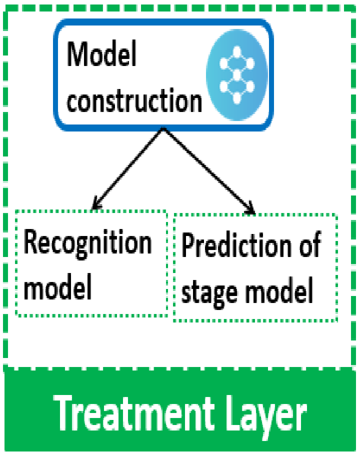
- Access data via API: storing data on the cloud enables the platform to access it from any location via internet-based API requests.

4.3.1.3 Treatment Layer:

It is responsible for the core analytical and decision-making processes. It involves constructing and utilizing DL and ML models for pest detection and prediction. so we mention here the model construction:

- Recognition Model: A CNN specifically trained to identify Tuta Absoluta from the sequence of images extracted from the video feed captured by the camera. This model analyzes the images in real-time and confirms the presence of the pest. The model is installed on the device.

- Prediction of Stage Model: A regression model that predicts the likelihood of Tuta Absoluta presence(each life stage of this insect) based on environmental data such as temperature and humidity. This model uses historical data and current environmental conditions to make predictions.



4.3.1.4 End User Layer:

It provides the interface for farmers and other stakeholders to interact with the system. It ensures that users are kept informed in real-time and can take appropriate actions based on the system's notifications and insights.

Mobile Application: A user-friendly app that receives real-time notifications from the system. It informs the farmer about detected pests, environmental conditions, and any actions taken by the system (tuta absoluta detection, elimination and prediction). So the farmer can monitor the greenhouse through the smartphone application.

In the end of this section, we say that This multi-layered architecture integrates advanced technologies and proactive pest management strategies to create a robust system for detecting and eliminating Tuta Absoluta in greenhouse environments. Each layer plays a crucial role in ensuring the system's efficiency, accuracy, and user-friendliness, ultimately helping farmers maintain healthy.

4.4 Tuta Absoluta detection system

In this section, we present the creation of our models. The model creation step consists of two operations. The first and most crucial operation is the preparation of the primary task, which involves classification and detection. The second additional operation is predicting the stage of the insect like showing in the below diagram 4.2:

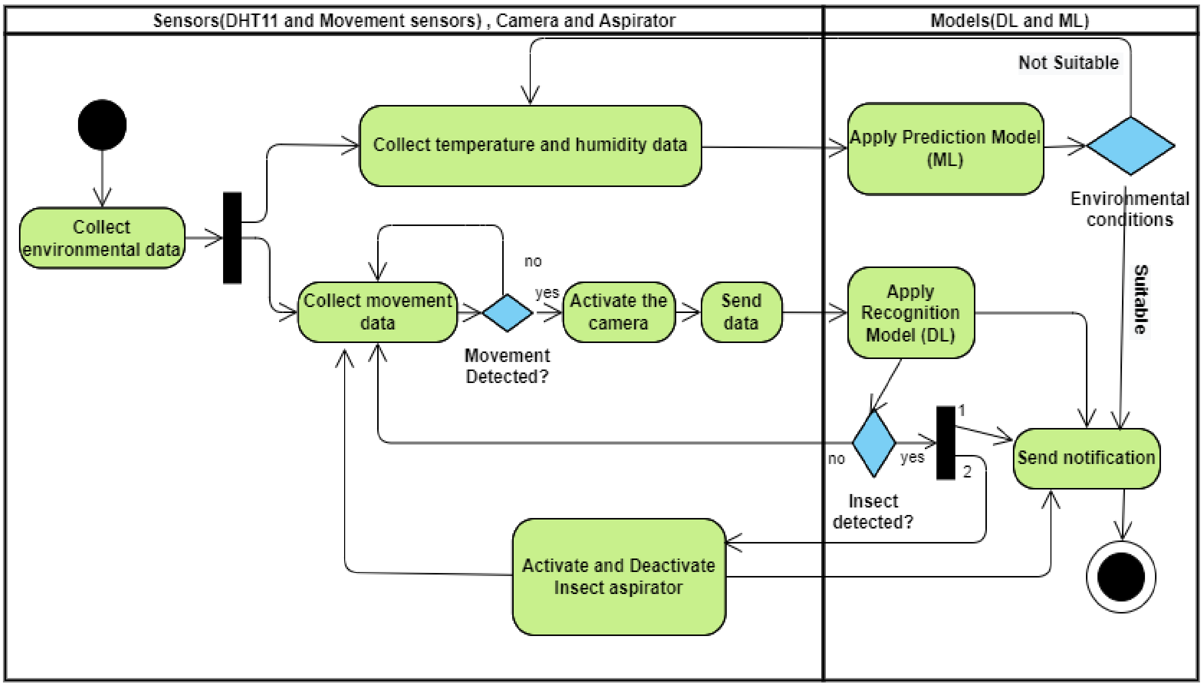

Sensors(DHT11 and Movement sensors) , Camera and Aspirator
Models(DL and ML)
Collect environmental data
Collect temperature and humidity data
Collect movement data
Movement Detected?
no
yes
Activate the camera
Send data
Activate and Deactivate Insect aspirator
Apply Prediction Model (ML)
Not Suitable
Environmental conditions
Suitable
Apply Recognition Model (DL)
Insect detected?
no
yes
1
2
Send notification

Figure 4.2: The full system for all the processes(detection, prediction and elimination)

4.4.1 Prediction of stage

In our effort to forecast the developmental stages of insects, particularly Tuta absoluta, we employ a rigorous methodology that combines Logistic Regression (LR) , Support Vector Machine (SVM) and support vector regression (SVR) models. Our goal is to predict the temporal manifestations of each stage of the insect's life cycle based on its age and the temperature conditions that are favorable for each stage's activity. By harnessing both LR ,SVM and SVR techniques, we strive to improve the precision and dependability of our predictions, enabling effective monitoring and control of the Tuta absoluta life cycle. The following flowchart (4.3) present the full process

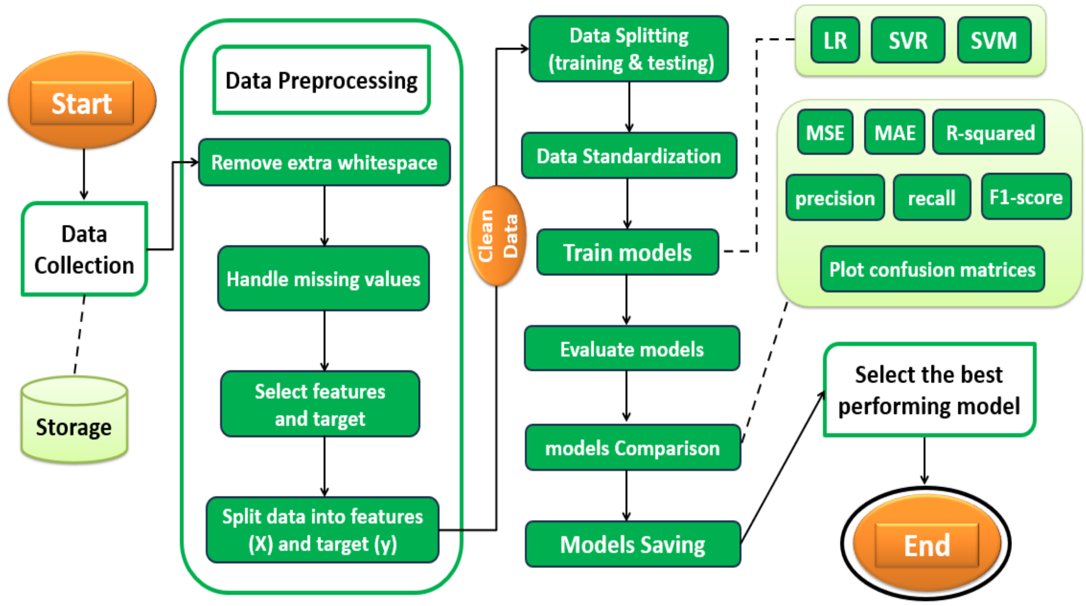
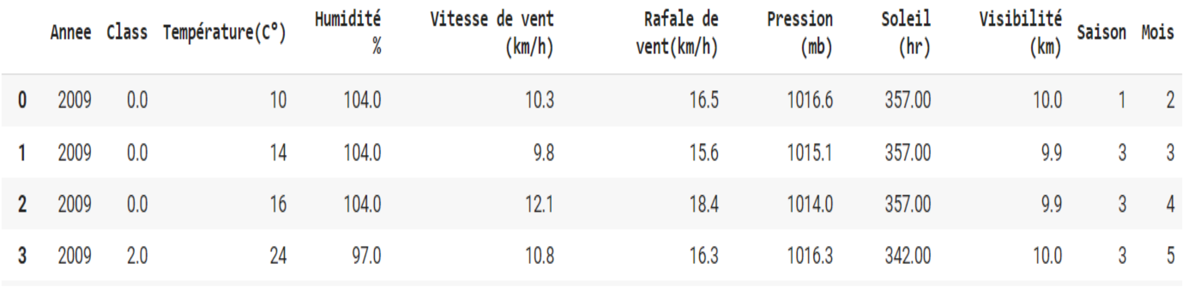


Figure 4.3: Flowchart of the prediction process

The flowchart illustrates a typical machine learning process for predicting the stage of an insect, commonly known as insect stage classification. Here's a breakdown of the steps:

- **Data Collection:** this involves gathering data about this insect, specifically focusing on features that can help distinguish different stages. So we gathered data for Biskra, including temperature (°C), humidity (%), wind speed (km/h), wind gust (km/h), pressure (mb), sunlight (hr), visibility (km), season, and month, spanning from 2009 to 2019. This data was categorized by month and season and synchronized with the growth phases of Tuta absoluta for precise modeling. Also we collect real time data to ensure the efficiency of our product. The collected information is structured and stored in a suitable format, CSV file, for ensuring easy access and manipulation during the subsequent steps (the training and testing of the LR, SVM, and SVR models).



	Annee	Class	Température(C°)	Humidité %	Vitesse de vent (km/h)	Rafale de vent(km/h)	Pression (mb)	Soleil (hr)	Visibilité (km)	Saison	Mois
0	2009	0.0	10	104.0	10.3	16.5	1016.6	357.00	10.0	1	2
1	2009	0.0	14	104.0	9.8	15.6	1015.1	357.00	9.9	3	3
2	2009	0.0	16	104.0	12.1	18.4	1014.0	357.00	9.9	3	4
3	2009	2.0	24	97.0	10.8	16.3	1016.3	342.00	10.0	3	5

Figure 4.4: meteorology dataset Biskra

- **Data Preprocessing** The gathered data is prepared by remove extra whitespace, addressing missing values, standardizing features, and encoding categorical variables if needed. This step ensures the data is clean and ready for model training.

```
# Remove whitespace characters from the column names
df.columns = df.columns.str.strip()

# Remove rows with missing values
df = df.dropna()
```

- Feature Engineering: Key features like insect stage (0, 1, 2, 3) is selected and modified to improve the models' predictive capabilities.

```
y = df['Class']  # Assuming 'class' is the target variable (0=egg, 1=
    larvae, 2=pupa, 3=adult)
```

- **Data Splitting** The collected dataset is divided into two parts:
  - Training Data: used to train the machine learning model.
  - Testing Data: used to evaluate the model's performance after training. This separation ensures that the model is not evaluated on the same data it was trained on, leading to a more robust assessment of its generalization ability.
- **Data Standardization:** the data features are scaled to ensure they are on a similar scale
- **Model Training:** our models: Logistic Regression (LR), Support Vector Machine (SVM), and Support Vector Regression (SVR) are trained on the training data. Hyperparameter tuning is conducted to optimize model performance.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
    =0.2, random_state=42)
```

```
from sklearn.svm import SVC
from sklearn.metrics import mean_squared_error, mean_absolute_error,
    r2_score, precision_score, recall_score, f1_score
# Create an SVM model
svm = SVC()
# Fit the model
svm.fit(X_train_scaled, y_train)
# Make predictions
y_pred_svm = svm.predict(X_test_scaled)
```

- **Model Evaluation:** after training, the model's performance is evaluated using the testing data using metrics such as accuracy, precision, recall, F1-score, Mean Squared Error, and Mean Absolute Error. These metrics aid in evaluating the models' performance and reliability. The evaluation metrics are defined as follows:

1. **Mean Squared Error (MSE)** is the average squared difference between predicted and actual values, emphasizing larger errors [81]. Here is the equation for MSE:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.1}$$

2. **Mean Absolute Error (MAE)** is the average absolute difference between predicted and actual values, treating all errors equally and being less sensitive to outliers [81]. Here is the equation for MAE:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.2}$$

3. **R-squared (R²)** is the proportion of the variance in the dependent variable that is explained by the independent variables in a regression model, ranging from 0 to 1 [82]. Here is the equation for R^2:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4.3}$$

   where:

   - n is the number of data points.
   - y_i is the actual value for the i-th data point.
   - $\hat{y}_i$ is the predicted value for the i-th data point.
   - $\bar{y}$ is the mean of the actual values.

4. **F1 Score:** is the harmonic mean of precision and recall, offering a balanced assessment [83]. It's calculated as:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

5. **Precision:** measures the accuracy of positive predictions [83]. It's calculated as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.5}$$

6. **Recall:** assesses the proportion of actual positives correctly identified by the model [83]. It's calculated as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.6}$$

- Note :

- TP (True Positive) is the count of instances correctly predicted as positive.
- FP (False Positive) is the count of instances incorrectly predicted as positive.
- FN (False Negative) is the count of instances incorrectly predicted as negative.

```
mse = mean_squared_error(y_test, y_pred_svm)
print("Mean Squared Error:", mse)
mae = mean_absolute_error(y_test, y_pred_svm)
print("Mean Absolute Error:", mae)
r2 = r2_score(y_test, y_pred_svm)
print("R-squared:", r2)
precision = precision_score(y_test, y_pred_svm, average='weighted')
print("Precision:", precision)
recall = recall_score(y_test, y_pred_svm, average='weighted')
print("Recall:", recall)
f1 = f1_score(y_test, y_pred_svm, average='weighted')
print("F1-score:", f1)
```

- **Models Saving:** the selected model is saved for future use in predicting the stage of new insects.
- Select the best performing model based on the evaluation metrics, the model with the best performance is chosen for deployment.
- **Forecasting:** the model is applied to predict the developmental stages of Tuta absoluta based on temperature and humidity data captured by the sensor. These forecasts provide insights into the insect's life cycle stage under specific environmental conditions.
- **Notification Automation:** In addition to stage prediction, we implement an automated notification system within a mobile application. Upon each prediction, the system sends alerts to farmers, enabling them to stay informed about the current stage of insect development and take proactive measures accordingly. This real-time communication ensures that farmers can promptly address pest-related challenges.

So our approach empowers farmers with remotly timely insights.

4.4.2 Recognition model

The recognition model is our CNN deep learning model. which identify tuta absoluta.

So in this section, we present the algorithm employed in creating our model. Then, we get in details of constructing our model, from data collection to the evaluation part. (See 4.5)

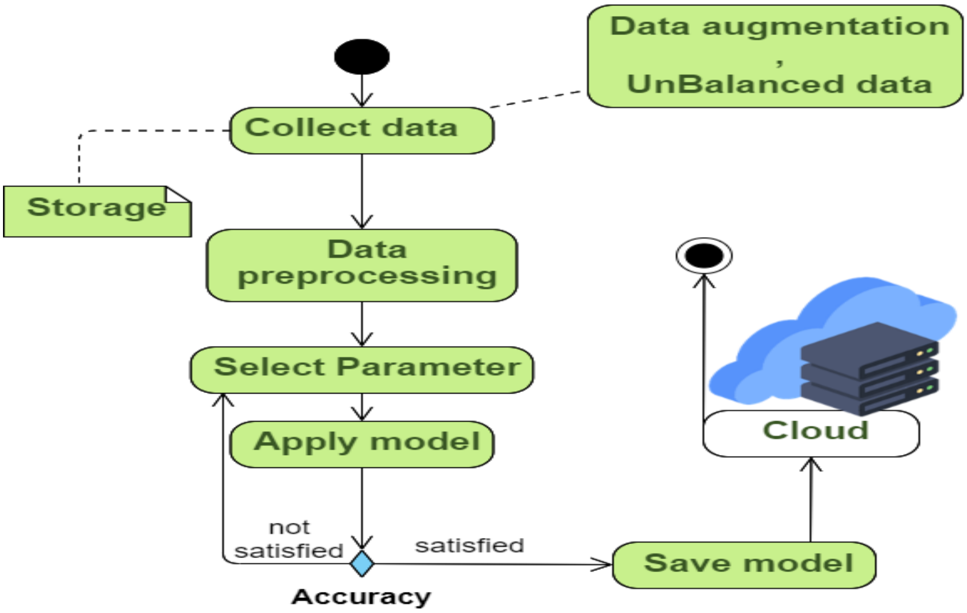


Figure 4.5: Tuta Absoluta detection system

Starting with convolutional neural networks (CNNs) Model, CNNs are deep learning models used for image tasks like classification, clustering, and object recognition. They have layers called convolutional layers, which use learnable filters to detect features in images. These filters are applied across the image to create activation maps, highlighting where features are found. CNNs also use down sampling layers to reduce the size of the data. Finally, CNNs include fully connected layers for classification, assigning labels to the features detected in the image [84].

The steps involved in creating our model are as follows:

1. Storage and Collection of Data: this is the initial step where we collect and store the dataset. First, we collected a dataset containing Tuta Absoluta images. Then, we organized them into separate directories based on the insect's life stages, these directories are named as follows: Tuta_absoluta_adult, Tuta_absoluta_egg, Tuta_absoluta_larva, and Tuta_absoluta_pupa. Each directory contains images corresponding to the respective life stage of the Tuta Absoluta insect. This dataset serves as the training and validation data for our **CNN** model. The model learns to classify these images based on their features, enabling it to distinguish between different stages of the Tuta Absoluta insect with high accuracy.

2. Data Preprocessing: this step involves preparing the data for the model. So prior to inputting the Tuta Absoluta insect data into the model, the dataset underwent several preprocessing steps.

   (a) Firstly, the images were resized to a fixed size of 200x200 pixels using OpenCV's 'cv2.resize' function. This step ensures that all images have the same dimensions, which is essential for training the model.

(b) Next, the pixel values of the resized images were normalized to a range of [0, 1]. This was achieved by dividing the pixel values by 255.0, which is the maximum pixel value for an 8 bit image. Normalization helps to scale the pixel values to a common range, making it easier for the model to learn from the data.

(c) Additionally, data augmentation techniques were applied to increase the size of the dataset and improve the model's robustness. This involved generating new images by applying random transformations. The following code snippet demonstrates the use of **ImageDataGenerator** to apply these transformations:

```
datagen = ImageDataGenerator(
    rotation_range=30,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    vertical_flip=True,
    brightness_range=[0.5, 1.5],
    fill_mode='nearest' )
```

The following chart4.6 provides Data with and without augmentation:

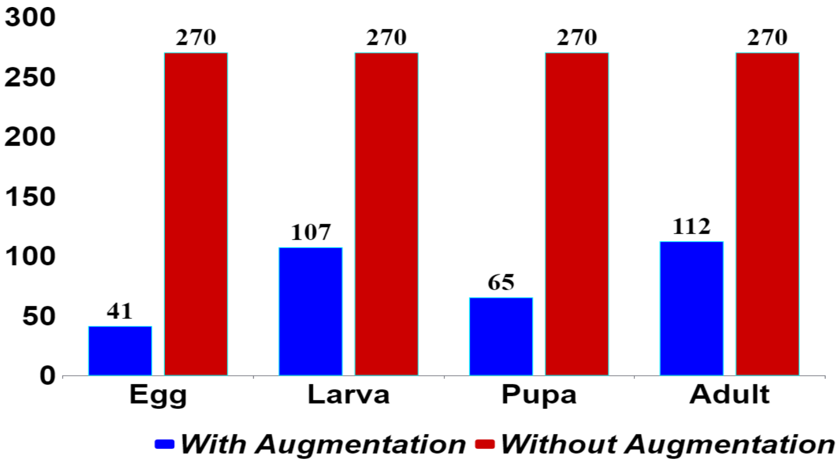
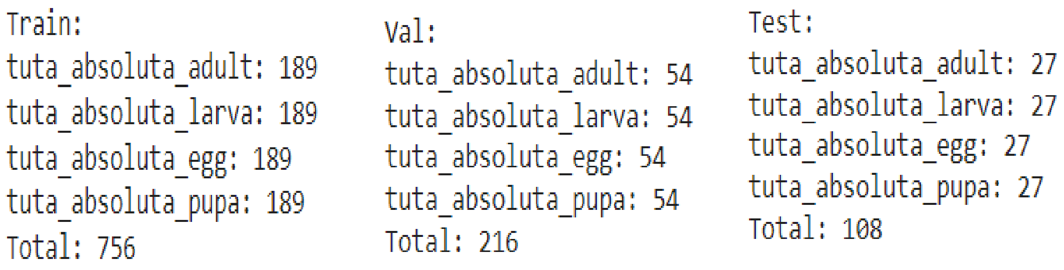


Figure 4.6: Data with/ without augmentation

The augmented images were then saved to a separate directory for further processing. And for the balancing, the data ensures that the model is trained on an equal number of samples from each class, which can help improve performance.

(d) After that, the dataset was split into training, validation, and test sets as shown in figure 4.7. The training set comprised 70% of the data, while the validation set comprised 20%, and the test set comprised 10%. This split ensures that the model is trained on a sufficient amount of data while also having separate datasets for evaluating its performance.



```
Train:
tuta_absoluta_adult: 189
tuta_absoluta_larva: 189
tuta_absoluta_egg: 189
tuta_absoluta_pupa: 189
Total: 756
```

```
Val:
tuta_absoluta_adult: 54
tuta_absoluta_larva: 54
tuta_absoluta_egg: 54
tuta_absoluta_pupa: 54
Total: 216
```

```
Test:
tuta_absoluta_adult: 27
tuta_absoluta_larva: 27
tuta_absoluta_egg: 27
tuta_absoluta_pupa: 27
Total: 108
```

Figure 4.7: Result of Splitting Data

3. Creating the model: the selection of an optimal model and architecture significantly influences the accuracy and performance of predictions. Our model architecture was based on the MobileNetV2 cnn, which is pretrained on the ImageNet dataset. This architecture was chosen for its efficiency and effectiveness in image classification tasks. The top layer of MobileNetV2 was removed, and a new classification head was added, consisting of a global average pooling layer followed by a dense layer with 128 units and ReLU activation, and a dropout layer with a dropout rate of 0.5. The output layer consisted of a dense layer with 6 units (corresponding to the number of classes) and softmax activation.

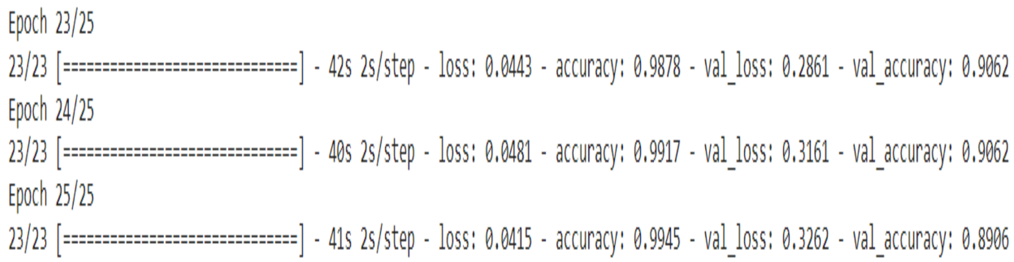
```
   x = GlobalAveragePooling2D()(base_model.output)
x = Dense(128, activation='relu')(x)
x = Dropout(0.5)(x)
output = Dense(4, activation='softmax')(x)
# Create the model
model = Model(inputs=base_model.input, outputs=output)
```

4. Compile the model: the model was compiled using the Adam optimizer, which is known for its efficiency in training deep neural networks. The categorical crossentropy loss function was chosen as it is suitable for multi-class classification problems. The model was configured to monitor the accuracy metric during training, which measures the percentage of correctly classified images.

```
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

5. Train and Apply the model: the Tuta Absoluta insect dataset was split into training and validation sets, with the training set used for model training and the validation set used for evaluating the model's performance. The training data was fed into the model in batches using an image data generator, which rescaled the pixel values to a range of [0, 1]. During training, the model iteratively adjusted its weights to minimize the defined loss function, aiming to improve its predictions. This process was repeated for 25 epochs (see figure 4.8), with the model's performance evaluated on the validation set after each epoch to monitor its progress and prevent overfitting.

```
history = model.fit(
   train_generator,
   steps_per_epoch=train_generator.samples // batch_size,
   epochs=25,  # Adjust the number of epochs as needed
   validation_data=val_generator,
   validation_steps=val_generator.samples // batch_size
```



```
Epoch 23/25
23/23 [==============================] - 42s 2s/step - loss: 0.0443 - accuracy: 0.9878 - val_loss: 0.2861 - val_accuracy: 0.9062
Epoch 24/25
23/23 [==============================] - 40s 2s/step - loss: 0.0481 - accuracy: 0.9917 - val_loss: 0.3161 - val_accuracy: 0.9062
Epoch 25/25
23/23 [==============================] - 41s 2s/step - loss: 0.0415 - accuracy: 0.9945 - val_loss: 0.3262 - val_accuracy: 0.8906
```

Figure 4.8: The ending of the training with the model

6. Condition of accuracy (No and Yes): This condition is related to accuracy. If it is good, we choose **yes**, and the model is ready. But if we don't achieve the accuracy we want, then **no**, and we go back to selecting parameters.

7. Save: After successfully the accuracy is high, we was saved trained model in the native Keras format using 'model.save(**'model.keras'**), and now we can save the model in the cloud.

8. Evaluate the model: The performance of the trained CNN model was assessed on the validation and test sets using Keras' `evaluate` method. This method computes the loss and accuracy of the model on the specified dataset, providing insights into how well the model generalizes to unseen data. Additionally, the F1 score, precision, and recall were calculated using the `classification_report` function from scikitlearn, which further evaluates the model's performance on the test set.

   We tested our CNN model on Tuta absoluta images to classify into four categories: Tuta absoluta adult, Tuta absoluta egg, Tuta absoluta larva and Tuta absoluta pupa. The accuracy of the most accurate model in the classification was measured.

   These ratios were calculated after training the CNN model with 25 epochs, indicating that the training data was traversed 25 times. Each time, the accuracy and loss of the model were calculated on the training data using validation data and labels to calculate validation measures. To monitor the progress of the model, and upon completion of the training, the model was evaluated on the test dataset.

4.4.3 Functional needs

This subsection consists of understanding the context of the proposed system. For our application, we define a set of business requirements that represent the actions of the system, confirmed by the following actions:

- **Create account Page :** Each Farmer can create account.
- **Login page: (Authentication)** The system (the application) must allow the user (Farmer) to enter their email and password to access the system. This process ensures the security of the system.
- **Collecting data in real time :** The system must be capable of collecting data in real time.
- **Real-Time identification of tuta absoluta:** The CNN deep learning model must be capable of detecting Tuta Absoluta in real-time.
- **Eliminator Activation:** Upon detection of Tuta Absoluta, the system should activate the Eliminator for pest elimination.
- **Prediction of the insect stage:** The ML model model must be capable of predict the stage of tuta Absoluta in real-time based on the collected data of temperature and humidity from the greenhouse.
- **Real-Time notifications:** Sending notifications to the mobile application which going to inform the farmer of the new events (detecting this insect, eliminate and the prediction).

4.4.4 Non-Functional needs

In this subsection we address the non-functional requirements. The key non-functional needs are as follows:

- **Accuracy:** Tuta Absoluta detection must achieve a high level of accuracy to minimize false positives and negatives.
- **Usability:** the application must be easy to use.
- **Security:** the application must respect the confidentiality of data.
- **Portability:** the application must be easy to installation, ,adaptation and interchangeability.

4.4.5 Use case diagram (system analysis model)

A use case diagram is a graphical representation that identifies the system and describes how it is used, as well as showing External features, user point of view and interactions with third parties [85].

To make it we must:

- Identify the actors in the system.
- Define use cases for each actor.

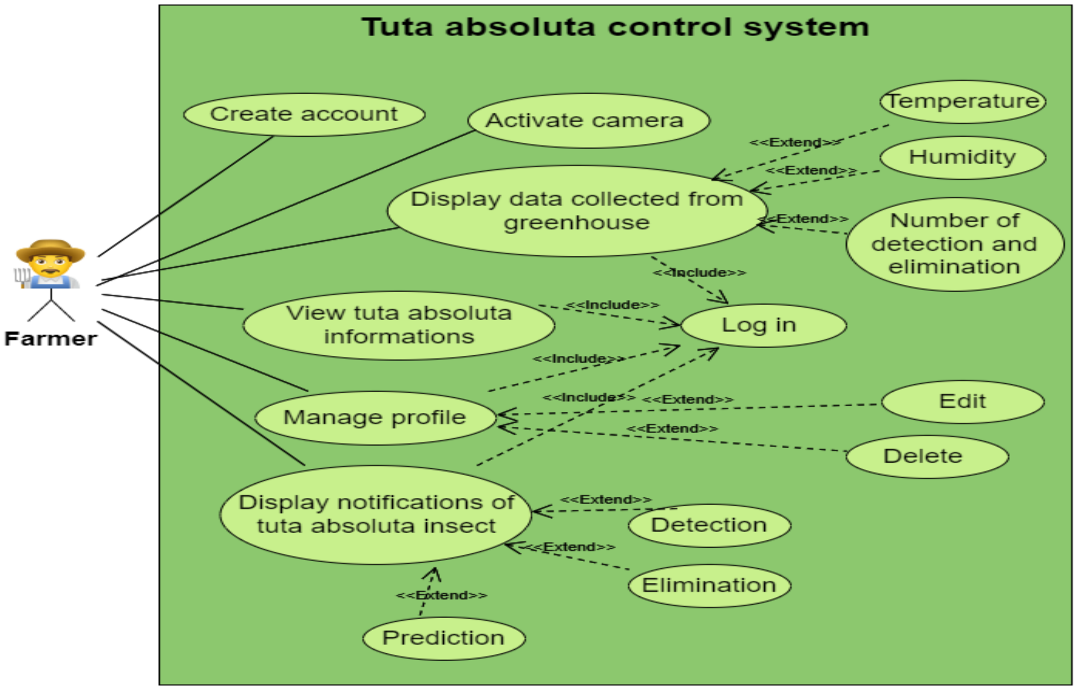


Figure 4.9: Use case diagram

4.4.5.1 Identification of actors

An **actor** is the user that represent a role played by an external person, process or thing that interacts with the system [85].

Within our system (our mobile application) which the the farmer, we identify one actors (See diagram 4.9):

farmer (primary actor): this actor represents the end-user, typically the greenhouse owner or manager.

4.4.5.2 Identification of use cases

The **use case** describes a function that a system performs to achieve the user's goal [85]. We present here an explanation for the use cases of the diagram 4.9:

- Create account: This case is mandatory to be able to access the system, in order to create an account, a user enters his information such first name, last name, e-mail, phone and other, or log in if he already has an account.
- Activate camera: The farmer can activate the device camera.
- Display data collected from greenhouse: The farmer can access data collected from sensors in the greenhouse, which can include temperature, humidity...etc.
- View the insect information: This use case allows a farmer to view information about this insect, such as its four stage and others.
- Manage profile: This use case allows a farmer to update their profile information, such as their name, contact information, or farm location.He can also delete old notifications(this use case allows a farmer to delete the old notifications).
- Display notifications of tuta absoluta: This use case allows the application to send a real time notifications to farmers about tuta absoluta, a tomato pest.

  1. Detection: This use case allows farmers to access information on how to detect tuta absoluta.
  2. Elimination: This use case allows farmers to access information on how to eliminate tuta absoluta.
  3. Prediction: this use case to predict the time of insect existence.

4.4.5.3 Sequence diagram of the ”Log in ” or ”sign up” scenario

The sequence diagram in figure 4.10 represents a possible login or sign up scenario:

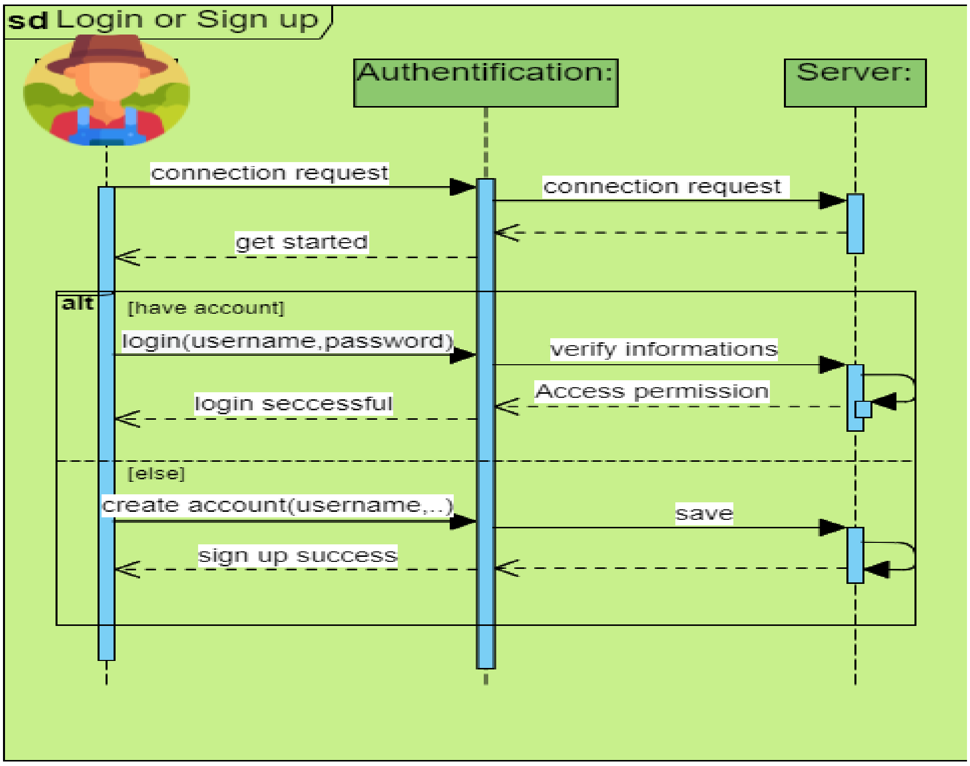


Figure 4.10: Sequence login diagram

- **Login or Sign Up**: The user sees a prompt asking them to either log in to an existing account or sign up for a new one.

- **[Have Account]**: This text likely appears when the user hovers their mouse over the login button. It clarifies that this button is for users who already have an account.

- **Login (Username, Password)**: If the user chooses to log in, they enter their username and password in the designated fields.

- **Verify Information**: Once the user enters their credentials, the server verifies the information to confirm it matches an existing account.

- **Login Successful / Access Permission**: If the username and password match an existing account, the user is granted access to the website.

- **[Else]**: If the user doesn't have an account, they can select this option to proceed with creating a new account.

- **Create Account (Username, etc.)**: The user enters their chosen username and other required information to create a new account.

- **Save**: Once the user enters their information, they click a button to save it and create their new account.

- **Sign Up Success**: If the account creation is successful, the user is granted access to the website.

4.4.5.4 Sequence diagram of the ”System work” scenario

The figure 4.11 represents the work of the system from Sensor Motion Detection to Farmer Notification.

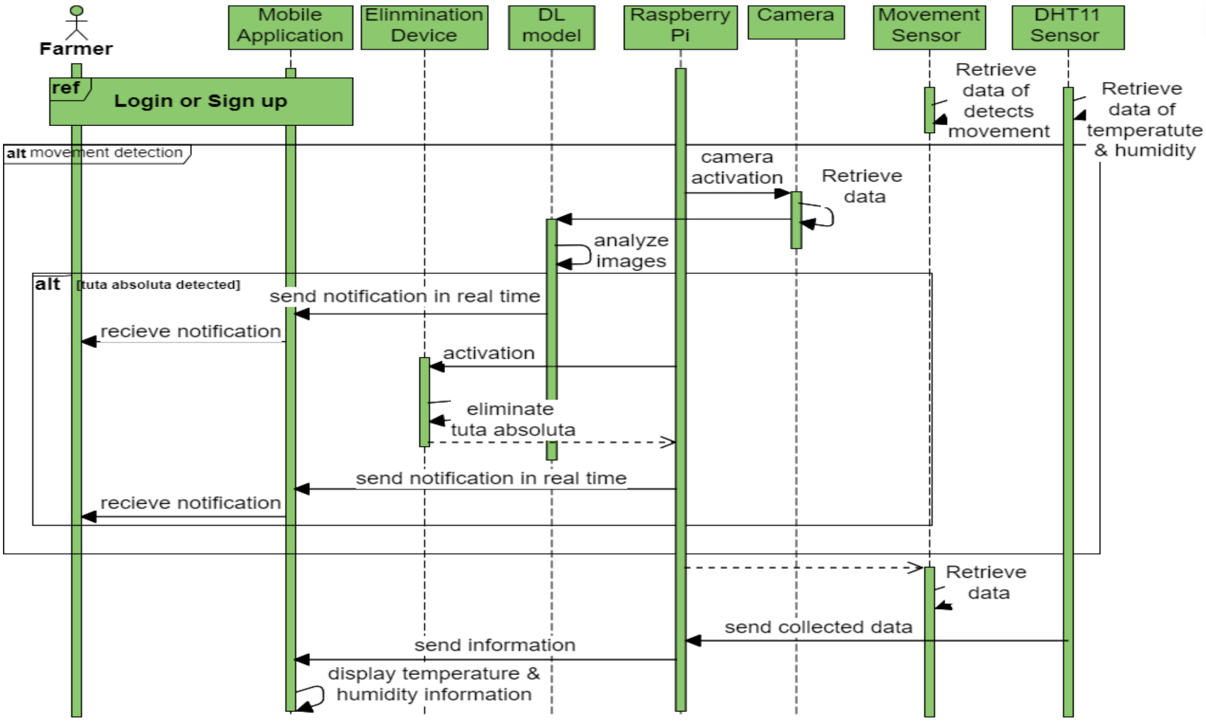


Figure 4.11: Sequence diagram of the system

1. **Movement Sensor:**
   - The Movement sensor senses any movement in the greenhouse.
   - Raspberry Pi receives the signal from the sensor indicating movement.
2. **Camera Activation:**
   - Upon receiving the signal from the ultrasonic sensor, Raspberry Pi activates the camera.
   - The camera capture a real-time video(retrieve data), which will be divided into a sequence of images. These images will then be analyzed by the CNN model to identify the presence of Tuta Absoluta.
3. **Insect Detection Using DL Model:**
   - The sequence of images are processed by the DL model.

- The DL model analyzes the images to detect the presence of the insects(tuta absoluta).

4. **Notification Sent if tuta absoluta Detected:**

    - If the DL model identifies Tuta absoluta in the images, the system send a notification to alert the application about the detection.

5. **Activation of Eliminator:**

    - After detecting this insect, the system activates the insect aspirator to eliminate tuta absoluta.

6. **Notification:**

    - If the verification process confirms successful elimination of Tuta absoluta, the system sends a notification.
    - Similarly, if any other significant event occurs during the process, such as a detection failure or system malfunction, a notification event is sent.

7. **Delivery to Mobile Application:**

    - The notification is delivered to the farmer's mobile application in real-time.
    - The mobile application presents the notification to the farmer, who can view it and take appropriate action based on the information provided.

Pseudocode for Tuta absoluta Detection and Elimination System

The following **pseudocode** for the sequence diagrams(full process):

Algorithm 1 Tuta absoluta Detection and Elimination System

```
//User Authentication
def user_authentication():
if user.has_account then
    Login
    login(user.username, user.password)
    verify_informations()
    access_permission()
    login_successful()
else
    Create account
    create_account(user.username, user.password)
    save_account_info()
    sign_up_success()
end if
//Detection and Elimination System
user_authentication()
Initialize devices and sensors
initialize_raspberry_pi()
initialize_ultrasonic_sensor()
initialize_DHT11_sensor()
initialize_camera()
initialize_aspirator()
initialize_mobile_app()
Main loop
while system_running do
    Retrieve data if movement detected
    movement_detected = ultrasonic_sensor.detect_movement()
    if movement_detected then
        Camera activation
        image = camera.active()
        Analyze images using DL model
        tuta_absoluta_detected = cnn_model.analyze_image(image)
        if tuta_absoluta_detected then
            Activate elimination device
            aspirator.activate()
            Notify farmer in real time
            mobile_app.send_notification(”Tuta absoluta detected and eliminated.”)
        end if
    end if
    Retrieve data if movement detected
    movement_detected = ultrasonic_sensor.detect_movement()
end while
```

4.5 Conclusion

In conclusion, this chapter has presented a detailed design and contribution for the Tuta Absoluta detection system. We have outlined the methodology and proposed architecture, emphasizing the work of each layer in achieving an efficient and effective detection process. The system's ability to recognize and predict stages of this insect has been discussed, along with the necessary functional and non-functional requirements. The diagrams ensures a comprehensive understanding of its operation such as the use case diagrams that provided a clear visualization of the system's interaction with users, and the sequence diagrams that present the full process of detection and elimination.

The next chapter will delve into the implementation details of our system and present the results obtained.

Chapter 5

Implementation and Results

5.1 Introduction

In this chapter, we describe the steps that we followed to develop and implement our system. Firstly, we provide a brief definitions of the programming languages such as Python and Java that we used to develop our system. As well as the tools like PyCharm and Android Studio, and libraries like Keras, OpenCV, TensorFlow, and NumPy, and frameworks that are important for our work. Additionally, we present the electronic equipments used to detect and eliminate harmful insects, specifically Tuta absoluta, in smart greenhouses, such as Arduino, ultrasonic sensors, aspirators, and cameras.

Then, we move on to the results of this work, including the outcomes of the CNN model and the mobile application. Next we explain all of these aspects in detail and showcase images of the proposed solution, which has now become a reality in the form of our prototype and application for farmers (device and app).

5.2 Implementation

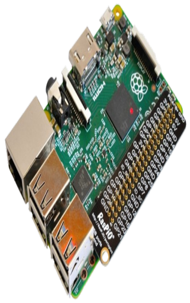
In this section, we discuss the hardware and software tools used in this project, including the frameworks, languages, and libraries.

5.2.1 Hardware tools - The electronic equipment -

In this sub-section, we describe the electronic devices used in our project.

5.2.1.1 Raspberry Pi 4 model B

The Raspberry Pi, introduced in 2012, is a compact, affordable, versatile, and education-focused computer board. It functions similarly to a traditional PC, requiring a keyboard for input, a display for output, and a power source. Most of its components, such as the central and graphics processing units, audio and communication hardware, and memory chip (ranging from 256 MB in Model A to 512 MB in Model B), are consolidated onto a single board [86]. To connect the Raspberry Pi to sensors like ultrasonic, temperature, and humidity sensors, electrical wires are used, with a breadboard serving as an intermediary linking tool.



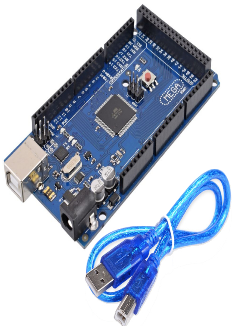
5.2.1.2 Raspberry Pi Camera

The Raspberry Pi Camera Module with Night Vision and Infrared (IR) Lens is a highly popular choice among Raspberry Pi enthusiasts. It features a 5-megapixel camera with a 222° extra-wide-angle fisheye lens (the field of view is 72° without the lens). This camera module is equipped with a high-quality OmniVision OV5647 CMOS image sensor and is compatible with all Raspberry Pi versions.

This module comes with a built-in IR CUT lens and two LED lights, allowing it to capture images and videos both during the day and at night or in low-light conditions. The LED lights automatically turn on in dark conditions to ensure clear and visible footage [87].

5.2.1.3 Arduino Uno

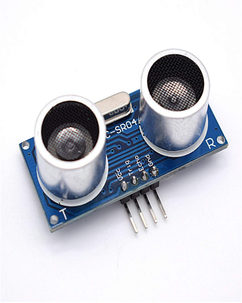
Arduino is an open-source company that offers microcontrollers and kits for creating digital devices. Their hardware is licensed under CC BY-SA, while the software is under LGPL or GPL, allowing anyone to manufacture Arduino boards and distribute the software. Arduino boards have digital and analog I/O pins for connecting expansion shields or other circuits. Programming is done using C and C++ (Embedded C) with the Arduino Programming Language API, influenced by Processing. Arduino provides an easy-to-use IDE and a Go-based command-line tool for development [88].



In our project, Arduino handled sensors outputs and transmitted them to Raspberry Pi for further processing.

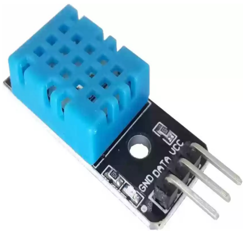
5.2.1.4 Ultrasonic sensor

The ultrasonic sensor is a device utilized for measuring the distance from the ground at specific points on a motor vehicle through the emission and reception of ultrasonic pulses. This sensor utilizes a constrained optimization method to capture reflected pulses, which are then analyzed by a threshold comparator for precise sub-wavelength detection. Experimental trials conducted with a 40 kHz piezoelectric-transducer based sensor demonstrated a standard uncertainty of 1 mm under stationary or low-speed conditions. Despite being able to operate at speeds of up to 30 m/s, the sensor exhibits increased uncertainty at higher speeds. Comprised of cost-effective components, the sensor possesses the ability to adapt to varying conditions autonomously, ensuring optimal performance. These characteristics render it a viable choice for inclusion as original equipment in numerous automotive applications [89].



5.2.1.5 The DHT11 sensor

The DHT11 is a digital sensor designed to measure both temperature and humidity levels. It delivers highly precise readings while maintaining long-term stability [90]. The sensor incorporates a resistive humidity measurement component and an NTC temperature measurement component, which are integrated with an 8-bit microcontroller to ensure rapid and reliable performance. Its calibration coefficient is stored in the OTP program memory, enabling accurate and consistent measurements. The DHT11 is compact, user-friendly, and suitable for a wide range of applications, with a signal transmission range extending up to 20 meters [91].



5.2.1.6 LCD Display

Liquid Crystal Displays (LCDs) are commonly used in electronics for visual output. They often have consistent pinouts across models, with eight data pins. The 4-pin mode uses D4-D7, while the 8-pin mode uses all eight pins. The 4-pin method is preferred for its simplicity. Other important pins include EN for enabling the display, RW for Read/Write mode, VSS and VDD for ground and +5V, and RS for selecting the register [92].

5.2.2 Frameworks, tools and libraries

5.2.2.1 Programming language

There are many programming languages.In our project, we used java and Python languages to program the application and the device, respectively.

1. **Python** a high-level, general-purpose programming language, was developed by Guido van Rossum and initially launched on February 20 [93]. Known for its simplicity and readability, Python is widely utilized in general-purpose programming, making it a favored option for both novice and seasoned programmers. Its applications span across various domains, including web development, software development, data analysis, machine learning, and scientific computing [94].
2. **Java** is a versatile programming language used for developing applications for the intranet and the World Wide Web (WWW). It follows the principle of ”Write Once, Run Anywhere” (WORA), meaning that Java code can be written on one platform and run on multiple platforms. Java is known for its network-friendly nature and platform independence, making it ideal for creating platform-independent and object-oriented applications. When Java source code is compiled, it is translated into bytecode, which can be executed on any platform with a Java Virtual Machine (JVM), ensuring platform independence [95].
3. **C++** is a versatile programming language that originated as an extension of the C language to incorporate object-oriented programming capabilities [96]. It was conceived by Danish computer scientist Bjarne Stroustrup at Bell Labs in 1985. The C++ Standard Library offers an extensive collection of pre-built functions and data structures, simplifying the development of intricate software solutions. Furthermore, C++ is recognized as a multi-paradigm language, accommodating procedural, object-oriented, and generic programming methodologies seamlessly [97].

4. **dart** is a client-optimized language designed for fast app development across platforms, aiming to enhance productivity with a flexible execution runtime. It supports core developer tasks like formatting, analyzing, and testing, and forms the foundation of Flutter. Dart's type-safe system includes sound null safety, and it features extensive core libraries and robust compiler technology for native and web platforms, ensuring efficient and high-quality production experiences. [98].

5.2.2.2 Libraries

In our project, we utilized several libraries to facilitate the implementation of our machine learning model and enhance its capabilities. These libraries played a crucial role in handling various aspects of the project, from data preprocessing to model training and evaluation. Some of the key libraries we used include:

1. **Open Source Computer Vision Library (OpenCv)** , commonly known as OpenCV, is a powerful open-source software library focused on computer vision, machine learning, and image processing. Originally developed by Intel, OpenCV is now maintained by a community of developers under the OpenCV Foundation [99].

2. **TensorFlow** is a freely available platform designed for machine learning through the use of data flow graphs. In these graphs, nodes signify mathematical operations, while the edges represent the multidimensional data arrays, or tensors, flowing between these operations. This adaptable framework enables the description of machine learning algorithms as a network of interconnected operations. What's more, these algorithms can be trained and run on various processing units such as GPUs, CPUs, and TPUs, and across different platforms, without the need to rewrite the code. This versatility allows TensorFlow to be used on a wide range of devices, from portable gadgets to desktop computers to powerful servers [100] .

3. **Keras** , an advanced deep learning API developed by Google using Python, aims to streamline the process of building neural networks. With its capability to work with various backend neural network computations, Keras stands out as a flexible solution for creating and implementing neural network models [101].

4. **NumPy** is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python [102].

5.2.2.3 Tools

In our project, we utilized various tools to streamline the development process and enhance our productivity. Some of the key tools we used include:

1. **Google Colab**

   Google Colab, short for Colaboratory, is A cloud-based platform for writing and executing Python code [103], It is designed to facilitate machine learning (ML) and data science tasks by providing a virtual environment with access to free GPU resources. [104]. Google Colab offers several benefits such as free access to GPUs, no setup required, collaborative editing, integration with Google Drive, support for popular libraries, and easy sharing of notebooks [103].

2. **Visual Studio**

   Visual Studio, created by Microsoft, is an integrated development environment (IDE) designed for building computer programs, including websites, web applications, web services, and mobile applications. It accommodates various programming languages and platforms, such as Windows API, Windows Forms, Windows Presentation Foundation (WPF), Windows Store, and Microsoft Silverlight [105].

3. **Android studio Emulator**

   The Android Emulator simulates Android devices on your computer, allowing you to test applications on various devices and API levels without needing physical devices. It offers flexibility with predefined configurations for different Android devices, high fidelity by simulating real device capabilities, and speed in data transfer. This makes the emulator an ideal choice for testing, though deploying to a physical device is also an option [106].

4. **Firebase**
   Firebase is a Google-backed platform that supports app development by providing tools to build and run modern, AI-powered experiences. It helps developers create high-quality apps that users love. Trusted by millions of businesses worldwide, Firebase simplifies the app development journey [107].
5. **Arduino ide** is a freely available software used for writing and uploading code to Arduino boards. It supports programming in C and C++ and offers a user-friendly code editor with features such as text editing, syntax highlighting, and automatic indentation. Additionally, the IDE includes a software library for commonly used input and output functions and utilizes avrdude for uploading code to Arduino boards. Overall, it provides a versatile and intuitive platform for programming Arduino boards [108].
6. **Raspberry Pi OS**
   Raspberry Pi Imager provides a simple and efficient method for installing Raspberry Pi OS and other operating systems onto a microSD card, ensuring it is ready for use with your Raspberry Pi [109].

5.3 Obtained results and discussion

In this section, we will discuss the results of our work, which are divided into two parts: model results and mobile application results

5.3.1 Model results

We used two types of models: CNN models for classifying Tuta absoluta images , and LR,SVR and SVM models for predicting the presence of Tuta absoluta based on temperature and humidity data

5.3.1.1 CNN Models Evaluation

The output of the model is a single continuous numerical value that represents the prediction of the Tuta absoluta insect's developmental stage (egg, larva, pupa, adult) based on the input image features.

1. **Confusion Matrix**

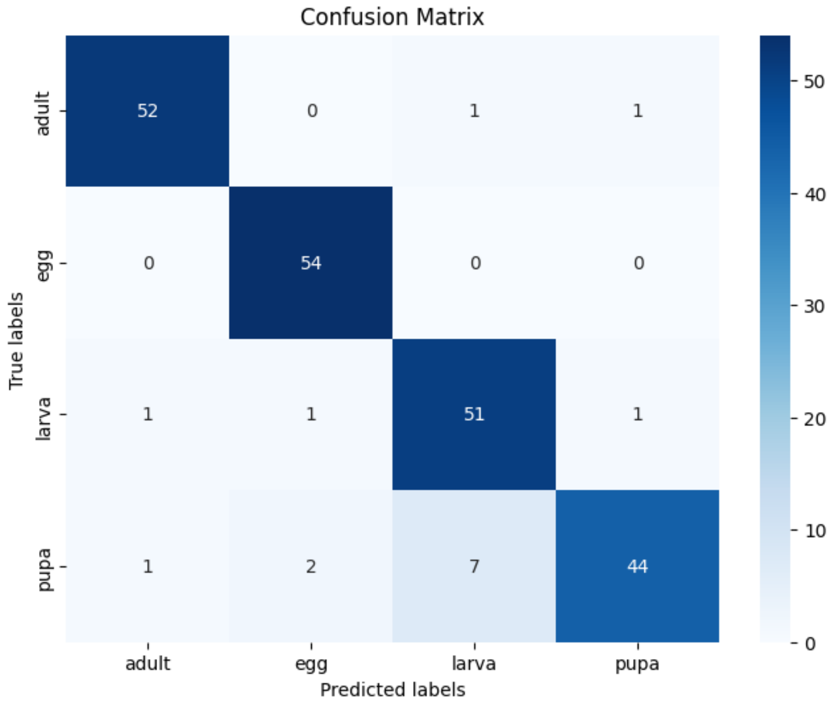


Figure 5.1: Confusion Matrix (CNN Model)

The confusion matrix visually summarizes the model's performance in classifying the four stages of Tuta absoluta. It shows the correct classifications on the diagonal and misclassifications off the diagonal. Each row corresponds to a true label, and each column corresponds to a predicted label.

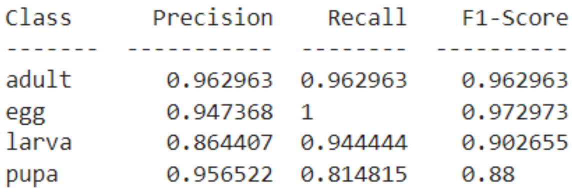
The confusion matrix in figure 5.1 has four rows and four columns, which correspond to the four classes in our dataset: ”egg”, ”larva”, ”pupa”, and ”adult”. The rows represent the true labels of the data points, while the columns represent the predicted labels.

Here's a breakdown of the matrix:

- The diagonal elements (from top left to bottom right) represent the number of data points that were correctly classified. For example, the first diagonal element (48) tells us that the model correctly classified 48 data points as ”egg”.
- The off-diagonal elements represent the number of data points that were misclassified. For example, the element in the fourth row and third column (3) tells us that the model misclassified 3 data points as ”pupa” when they were actually ”larva”.

2. **Performance Metrics**

- After plotting the confusion matrix, we calculated the F1 score, recall, and precision of each classe. Here are the results:



```
Class      Precision    Recall    F1-Score
-------  -----------  --------  ----------
adult       0.962963  0.962963    0.962963
egg         0.947368  1           0.972973
larva       0.864407  0.944444    0.902655
pupa        0.956522  0.814815    0.88
```

Figure 5.2: Classification report

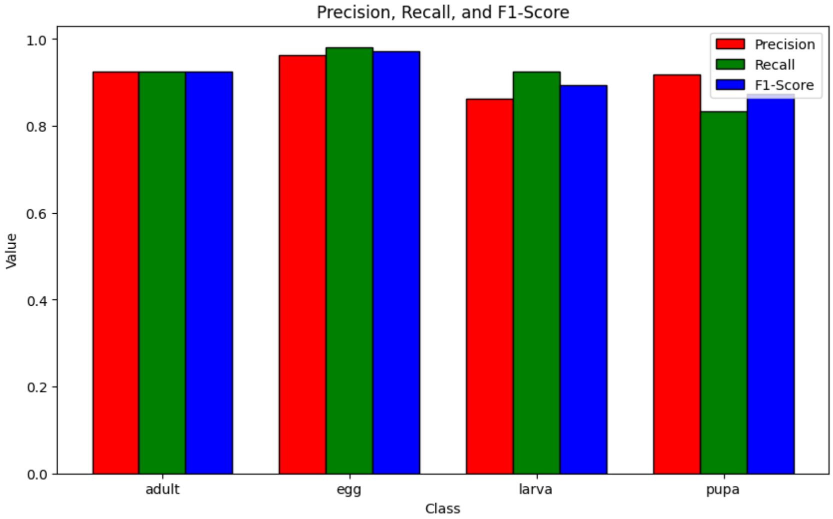


Figure 5.3: Performance Metrics - Precision, Recall, and F1-Score by Class

- Evaluation of Classification Accuracy and Loss. The following table (Table 5.1) displays the classification accuracy (Acc) and loss metrics of the Convolutional Neural Network (CNN) model employed in this study, specifically utilizing datasets comprising images of Tuta absoluta insects. This analysis provides insights into the performance of the CNN model in accurately classifying and evaluating loss on the Tuta absoluta insect image datasets.

Metric	**Value**
Train Acc	100%
Train Loss	0.0049
Val Acc	93.75%
Val Loss	0.2166

Table 5.1: Results of CNN models

The graph depicts the progression of accuracy and loss measures for a Convolutional Neural Network (CNN) model.

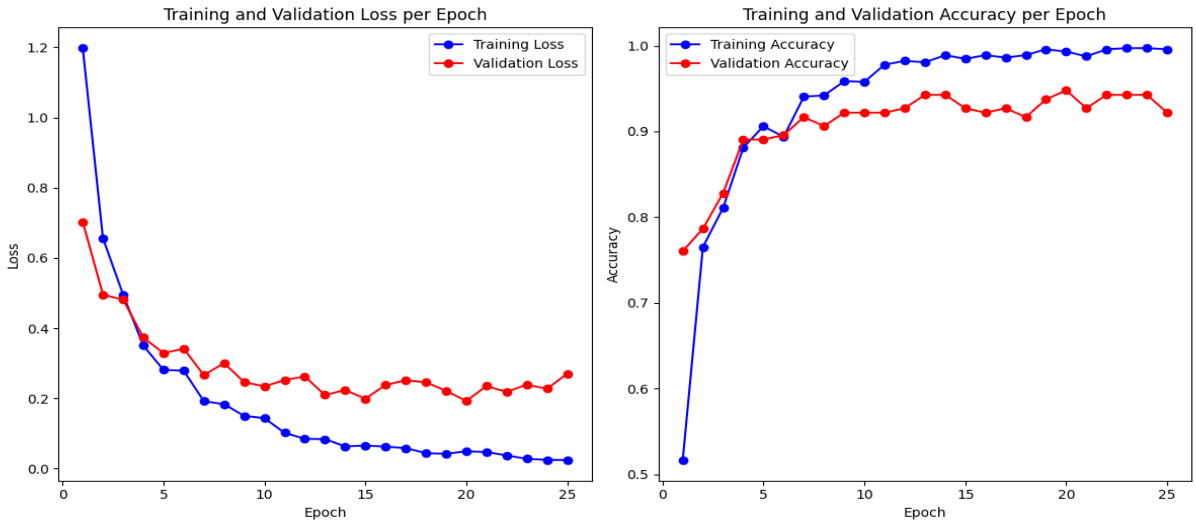


Figure 5.4: Accuracy and loss Curves of CNN model

The 5.4 represents the performance of a CNN model during training and validation over a number of epochs. An epoch is one complete pass through the entire training dataset.

The training loss and accuracy tell us how well the model is fitting the training data. At the beginning of training, the model is making many mistakes, as evidenced by the high training loss and low training accuracy. However, as training progresses, the model becomes more accurate and the training loss decreases.

The validation loss and accuracy, on the other hand, tell us how well the model is generalizing to new data. Ideally, we want the validation loss and accuracy to improve as training progresses, indicating that the model is not only fitting the training data well but is also able to generalize to new data.

In this case, we can see that both the training and validation accuracy are improving as training progresses, indicating that the model is learning to recognize patterns in the data. However, the validation accuracy is consistently lower than the training accuracy, which suggests that the model may be overfitting to the training data. Overfitting occurs when a model is too complex and fits the training data too closely, resulting in poor generalization to new data.

To address overfitting, we can try techniques such as regularization, early stopping, or reducing the complexity of the model. It's also possible that the model is underfitting, in which case we may need to increase the complexity of the model or train for more epochs.

3. **Collecting Data in Real Time**

   After training the model and designing the confusion matrix, we tested the model in real time to assess its performance on newly captured images. The real-time testing involved capturing images of Tuta absoluta insects and using the trained model to classify them. The model demonstrated varying degrees of confidence in its classifications, as shown in Figure 5.10. The results indicate that the model is capable of classifying Tuta absoluta insects in real time, with some images classified with high confidence while others may require further refinement of the model.

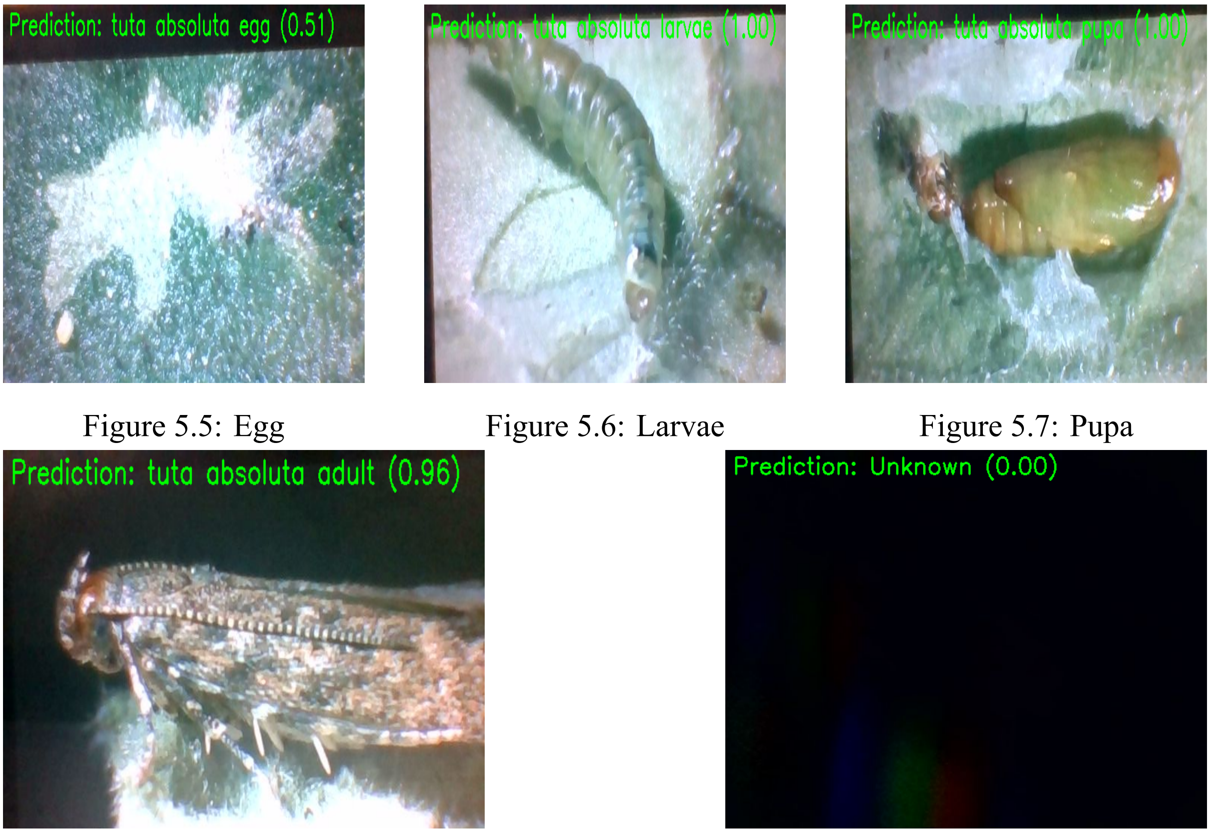

Prediction: tuta absoluta egg (0.51)
Prediction: tuta absoluta larvae (1.00)
Prediction: tuta absoluta pupa (1.00)
Prediction: tuta absoluta adult (0.96)
Prediction: Unknown (0.00)

Figure 5.5: Egg

Figure 5.6: Larvae

Figure 5.7: Pupa

Figure 5.8: Adult

Figure 5.9: Unknown

Figure 5.10: Results of real-time classification of Tuta absoluta insects.

If the model does not recognize the insect, it displays **"Unknown"** in the prediction. This allows for transparency in the model's limitations and ensures that unidentified images are not misclassified.

5.3.1.2 LR , SVM and SVR Models Evaluation

Based on environmental variables including temperature and humidity, we use Logistic Regression (LR), Support Vector Machine (SVM), and Support Vector Regression (SVR) models in our work to forecast the developmental stages of the Tuta absoluta insect. Every model offers a distinct method for regression and classification, offering insightful information for many facets of our predictive analysis.

- Logistic Regression (LR) is used to classify the insect's developmental stages into discrete categories (egg, larva, pupa, adult). The performance of the LR model is evaluated using accuracy, precision, recall, and F1-score metrics, .
- The SVM model is also utilized for classification purposes, aiming to enhance the precision and reliability of stage predictions. We assess the SVM model's performance using similar metrics as the LR model, such as accuracy, precision, recall, and F1-score.
- Support Vector Regression (SVR) is employed to predict the exact age and developmental stage of the insect based on continuous environmental variables. The performance of the SVR model is measured using Mean Squared Error (MSE) and Mean Absolute Error (MAE).

1. **Performance Metrics**

   after train the models , We computed the precision, recall, F1 score, Mean Squared Error, Mean Absolute Error, and R-squared. The outcomes are as follows:

Metric	**LR**	**SVR**	**SVM**
Mean Squared Error	0.25	0.12	0.21
Mean Absolute Error	0.25	0.27	0.21
R-squared	0.83	0.92	0.86
Precision	0.74	-	0.85
Recall	0.75	-	0.79
F1-score	0.73	-	0.76
Accuracy	0.91	0.94	0.85

Table 5.2: Performance Metrics for LR, SVR, and SVM Models

The plot below illustrates the performance metrics of (LR, SVR and SVM) models, evaluating them on (MSE), (MAE), R-squared, Precision, Recall, F1-score, and Accuracy.

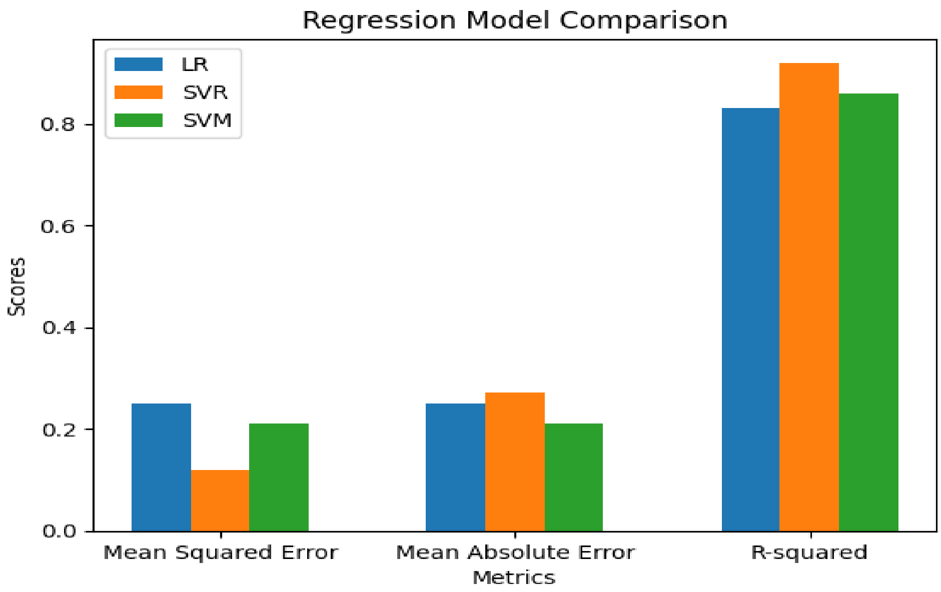


Figure 5.11: Comparison MAE, MSE and R2 of SVR, LR, and SVM

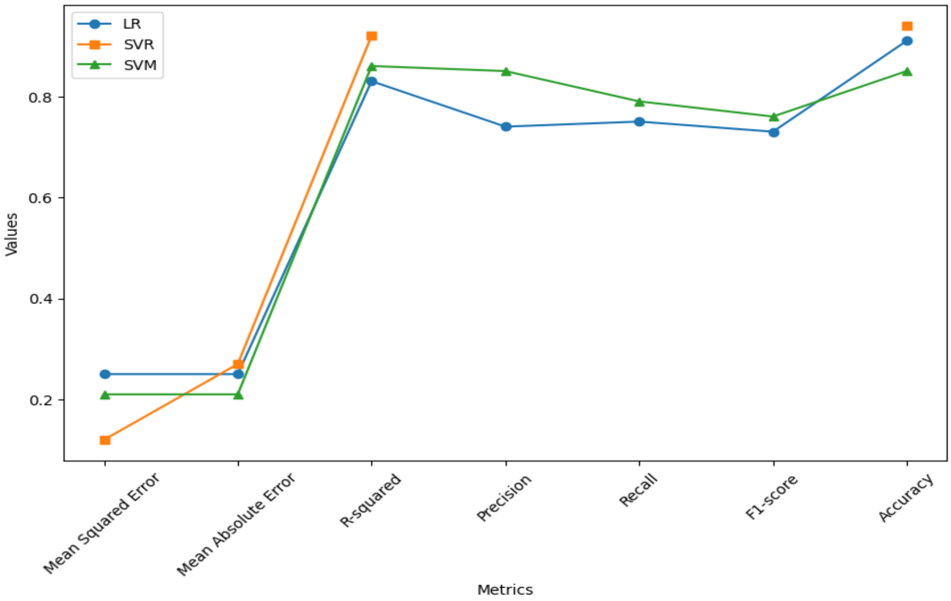


Figure 5.12: Performance Metrics for (LR,SVR and SVM) Models

5.4 Prototype Showcase: Detection and Elimination Mechanism

The figure 5.13 showcase the prototype created for detecting and eliminating Tuta absoluta . This device integrates several components, including an Arduino, Raspberry Pi, ultrasonic sensor, DHT11 sensor, Raspberry Pi camera,(Green and yellow) Leds and hormone (to attract insects) and a CD motor (aspirator), all working in harmony to identify, predict, and respond to the presence of adult insects.

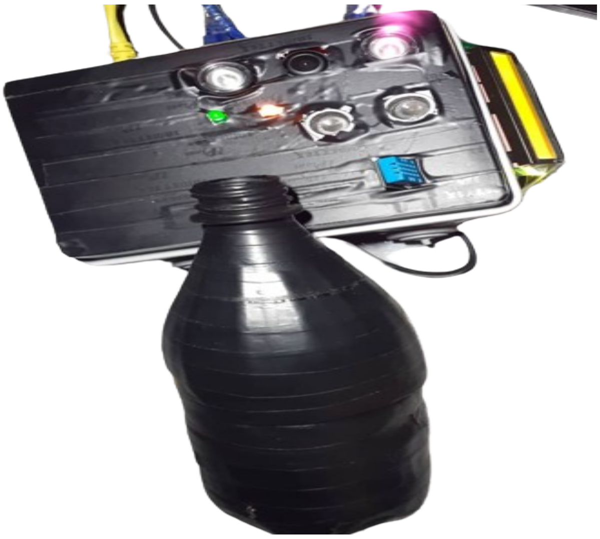
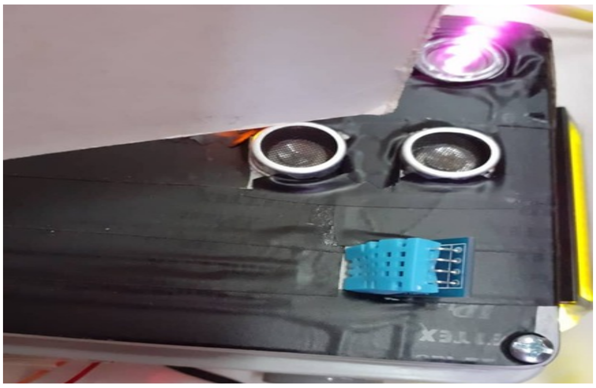
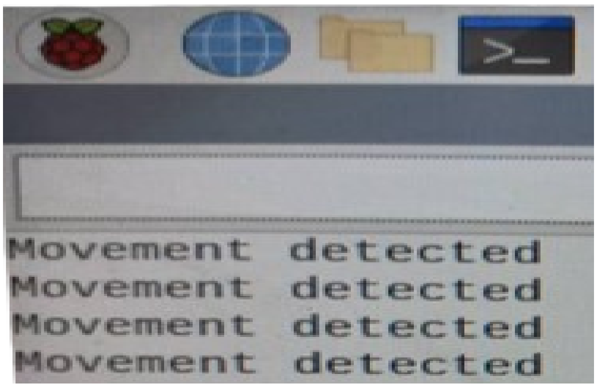


Figure 5.13: A picture of our device

The prototype works by utilizing the ultrasonic sensor to detect movement. When any object is moved near this sensor, as shown in Figure 5.14a, the Arduino code print the result as follow in the figure 5.14b



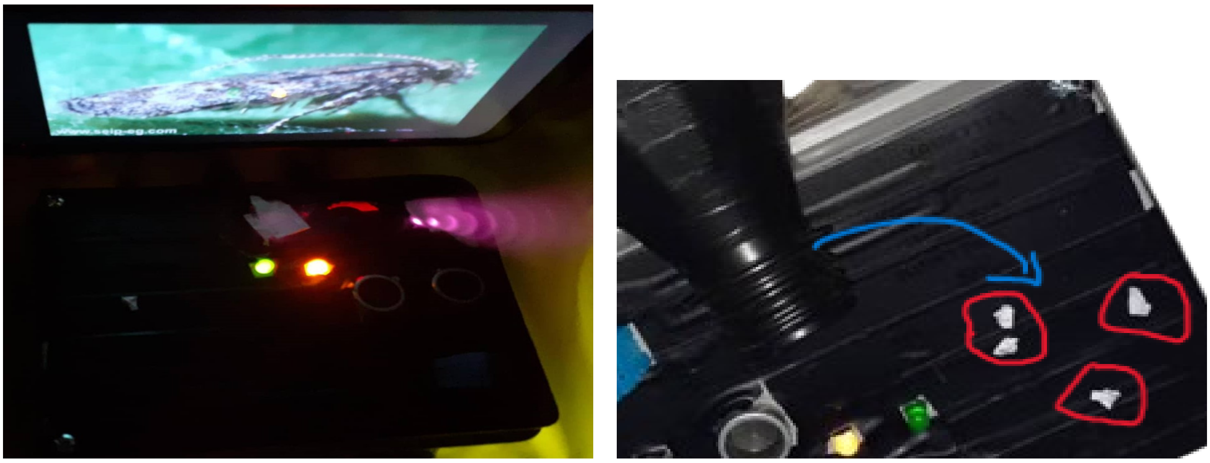
(a) Moving an object near the ultrasonic sensor



```
Movement detected
Movement detected
Movement detected
Movement detected
```

(b) Arduino code results

which triggers the Raspberry Pi camera to capture images as shown in figure 5.15a. These images are then classified using a CNN model. If an adult Tuta absoluta is detected, the device activates the CD motor (aspirator) to initiate an elimination mechanism shown in figure 5.15b. Additionally, the DHT11 sensor is used to predict the insect's presence based on environmental conditions and print the result in LCD Display as follow in figure 5.16 .



(a) Test the CNN model

(b) Swallowing the insect

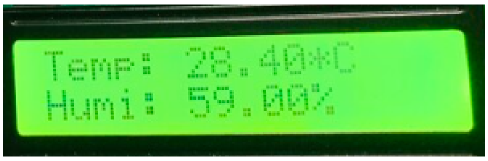
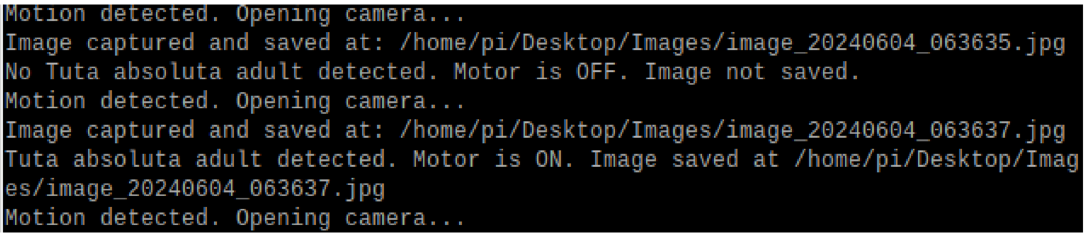


Figure 5.16: Result of DHT11 Sensor



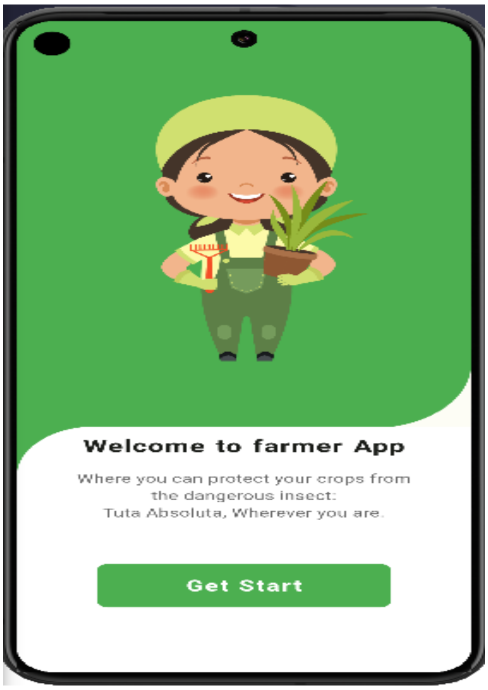
```
Motion detected. Opening camera...
Image captured and saved at: /home/pi/Desktop/Images/image_20240604_063635.jpg
No Tuta absoluta adult detected. Motor is OFF. Image not saved.
Motion detected. Opening camera...
Image captured and saved at: /home/pi/Desktop/Images/image_20240604_063637.jpg
Tuta absoluta adult detected. Motor is ON. Image saved at /home/pi/Desktop/Imag
es/image_20240604_063637.jpg
Motion detected. Opening camera...
```

Figure 5.17: result of python code on raspberry pi

5.4.1 Mobile application interfaces

In this subsection, we introduce our mobile application with user-friendly design, showcasing its features, its various functionalities and how it addresses user needs effectively.

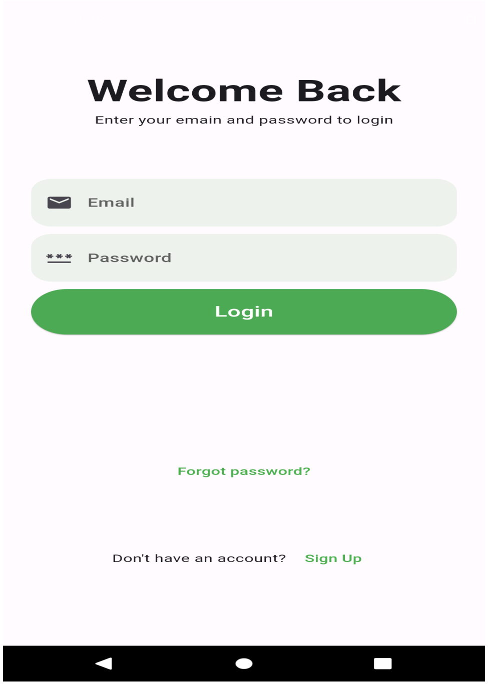
1. The first two pages are the welcome and login pages. The login page asks the user to enter their email and password to log in. It also provides links for users like who need to create an account.



Welcome to farmer App

Where you can protect your crops from the dangerous insect: Tuta Absoluta, Wherever you are.

Figure 5.18: Welcome page



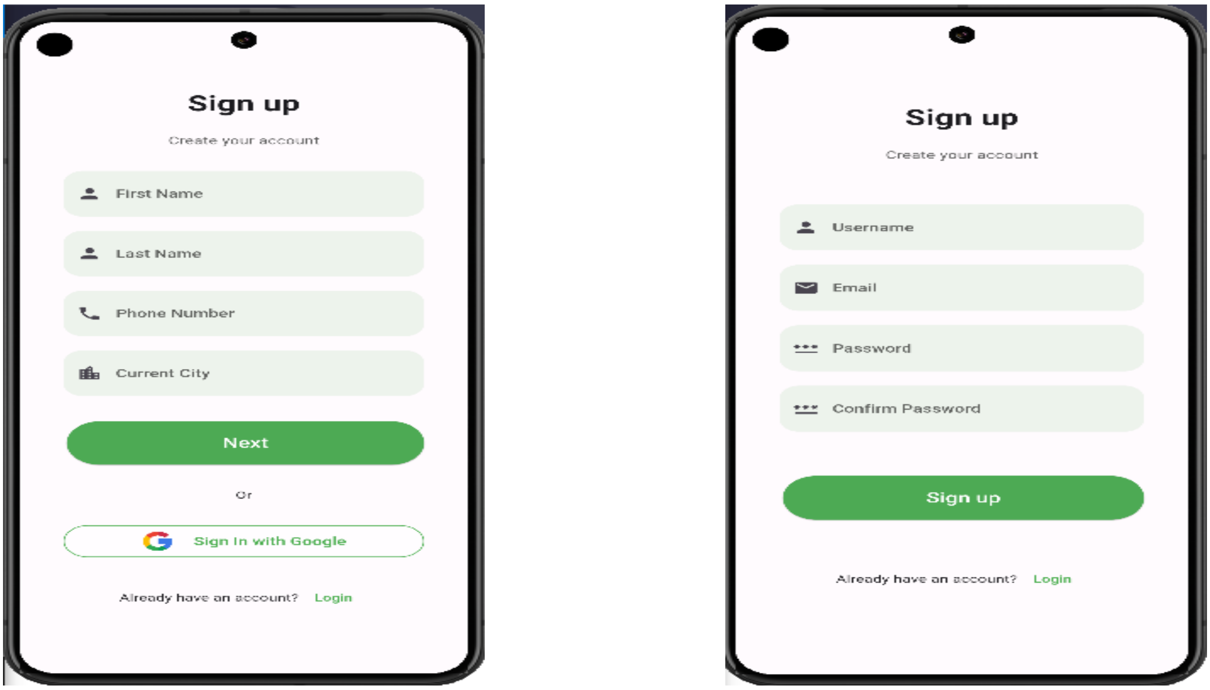
Welcome Back

Enter your emain and password to login

Don't have an account? Sign Up

Figure 5.19: Login page

2. For the sign up page, The users enter his personal details like first name, last name, phone number and current city. Then, he taps ”Next” to proceed to the next step which involves creating a password and entering an email address.



Sign up

Create your account

Already have an account? Login

Sign up

Create your account

Already have an account? Login

Figure 5.20: Sign up pages

3. The home page contains dashboard of the application services such as notifications, profile, viewing data collected, and camera activation. Users can easily access these services

from the homepage of the app. For the information page, the user will find guide and advices about this insect and the traditional control management.

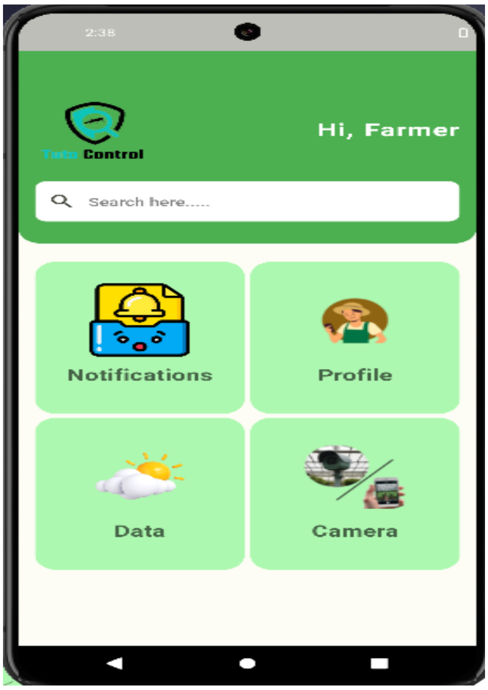
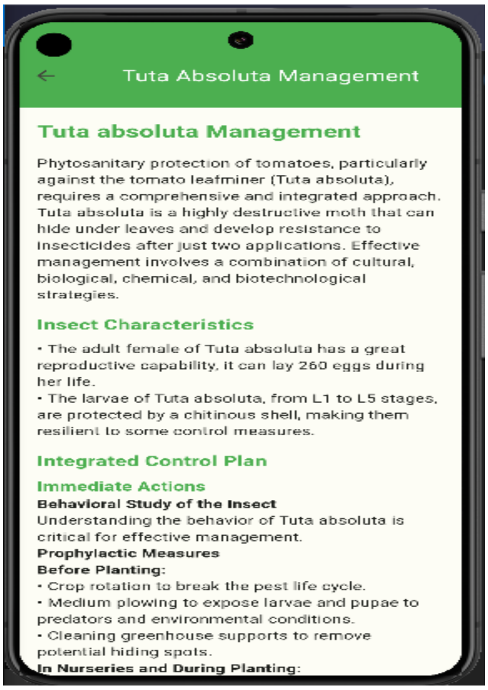


Figure 5.21: Home page



Tuta absoluta Management

Phytosanitary protection of tomatoes, particularly against the tomato leafminer (Tuta absoluta), requires a comprehensive and integrated approach. Tuta absoluta is a highly destructive moth that can hide under leaves and develop resistance to insecticides after just two applications. Effective management involves a combination of cultural, biological, chemical, and biotechnological strategies.

Insect Characteristics

- The adult female of Tuta absoluta has a great reproductive capability, it can lay 260 eggs during her life.
- The larvae of Tuta absoluta, from L1 to L5 stages, are protected by a chitinous shell, making them resilient to some control measures.

Integrated Control Plan

Immediate Actions

Behavioral Study of the Insect

Understanding the behavior of Tuta absoluta is critical for effective management.

Prophylactic Measures

Before Planting:

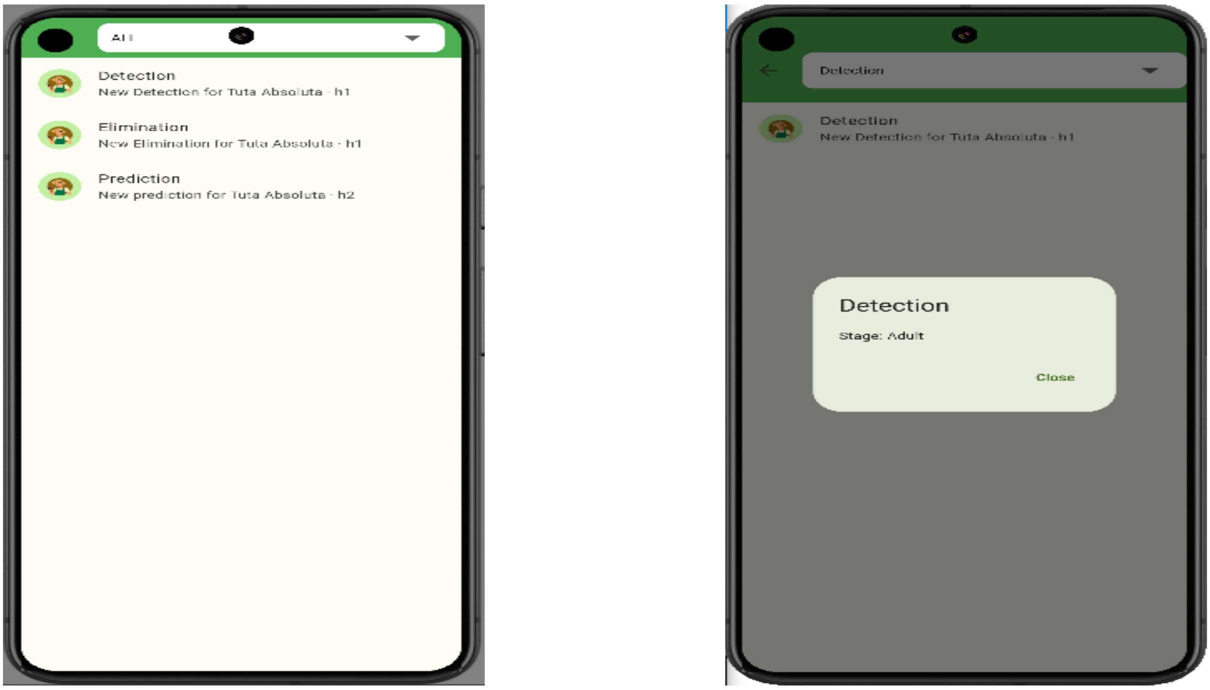
- Crop rotation to break the pest life cycle.
- Medium plowing to expose larvae and pupae to predators and environmental conditions.
- Cleaning greenhouse supports to remove potential hiding spots.

In Nurseries and During Planting:

Figure 5.22: Information page

4. Then notifications, where the user will receive different notifications in real time. This page displays a list of notifications that help users stay informed about the latest developments and actions. The notifications can be organized to categories, which are:

   - All: This section shows all notifications, regardless of category.
   - Detection: This section shows notifications related to the insect detection event.
   - Elimination: This section shows notifications related to the insect elimination event.
   - Prediction: This section shows notifications related to prediction events. This page likely belongs to a system that monitors and manages pests or other agricultural issues. The notifications help users stay informed about the latest developments and actions needed.



Detection
New Detection for Tuta Absoluta - h1

Elimination
New Elimination for Tuta Absoluta - h1

Prediction
New prediction for Tuta Absoluta - h2

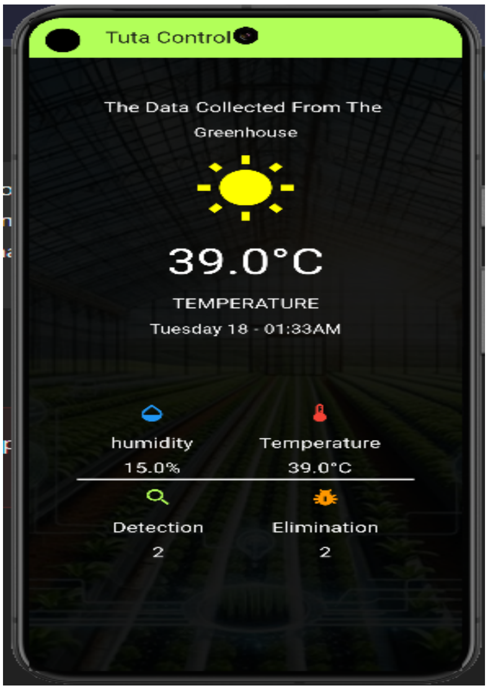
Detection
New Detection for Tuta Absoluta - h1

Detection
Stage: Adult

Figure 5.23: Notification page

5. The data page displays real-time data collected from a greenhouse. This page shows the current humidity and temperature in the greenhouse, along with the number of pests detected and the number of pests eliminated.

   So the data page provides valuable insights into the conditions of the greenhouse, helping farmers to optimize growing conditions and protect their crops. The data can be used to adjust other measures.



Tuta Control

The Data Collected From The Greenhouse

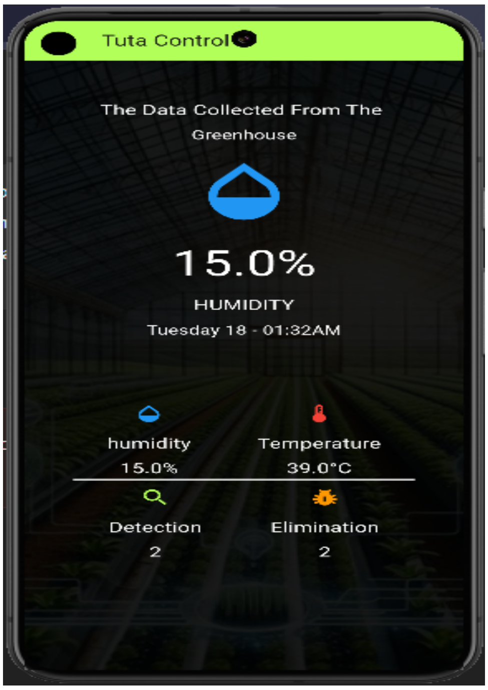
39.0°C

TEMPERATURE

Tuesday 18 - 01:33AM

humidity	Temperature
15.0%	39.0°C
Detection	Elimination
2	2

Figure 5.24: Temperature data



Tuta Control

The Data Collected From The Greenhouse

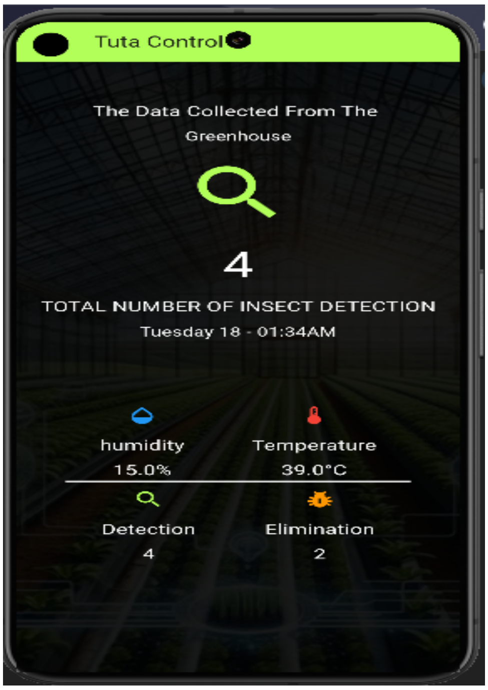
15.0%

HUMIDITY

Tuesday 18 - 01:32AM

humidity	Temperature
15.0%	39.0°C
Detection	Elimination
2	2

Figure 5.25: Humidity data



Tuta Control

The Data Collected From The Greenhouse

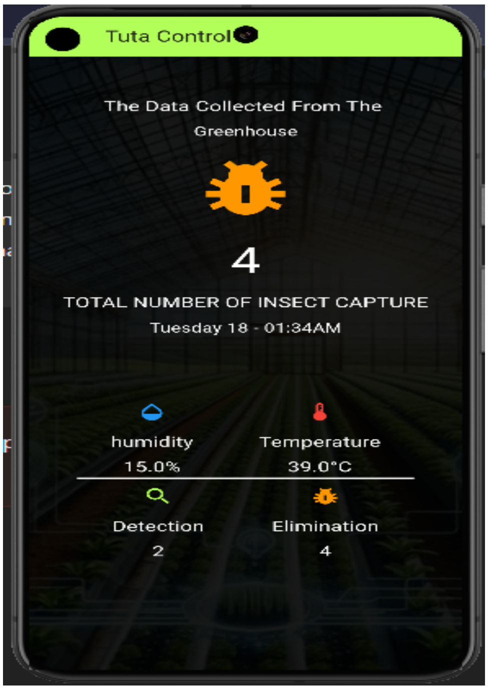
4

TOTAL NUMBER OF INSECT DETECTION

Tuesday 18 - 01:34AM

humidity	Temperature
15.0%	39.0°C
Detection	Elimination
4	2

Figure 5.26: Number of detection



Tuta Control

The Data Collected From The Greenhouse

4

TOTAL NUMBER OF INSECT CAPTURE

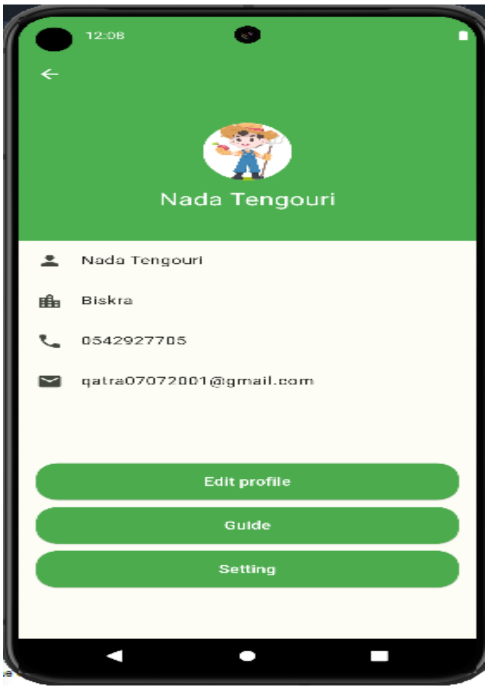
Tuesday 18 - 01:34AM

humidity	Temperature
15.0%	39.0°C
Detection	Elimination
2	4

Figure 5.27: Number of elimination

Figure 5.28: Collected Data page

6. Here two pages: the camera activation page, where the user can activate or deactivate the device camera using just his mobile phone. and the profile page includes details like current city, phone number, email and also has buttons for editing profile, guides(helpful information) and settings.



Nada Tengouri

Biskra

0542927705

qatra07072001@gmail.com

Figure 5.29: Profile page

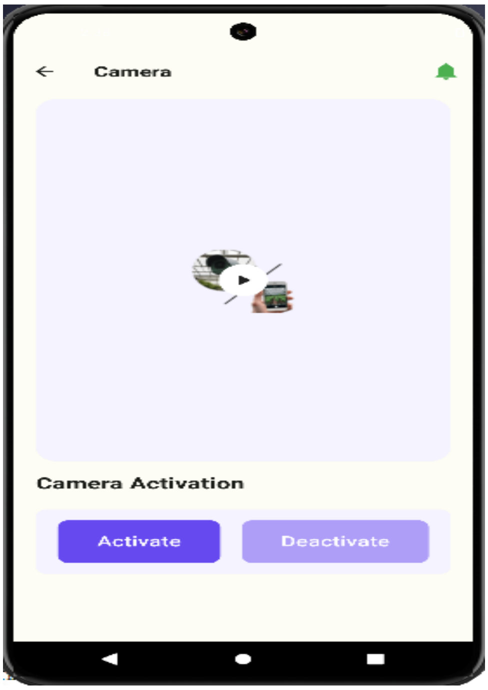
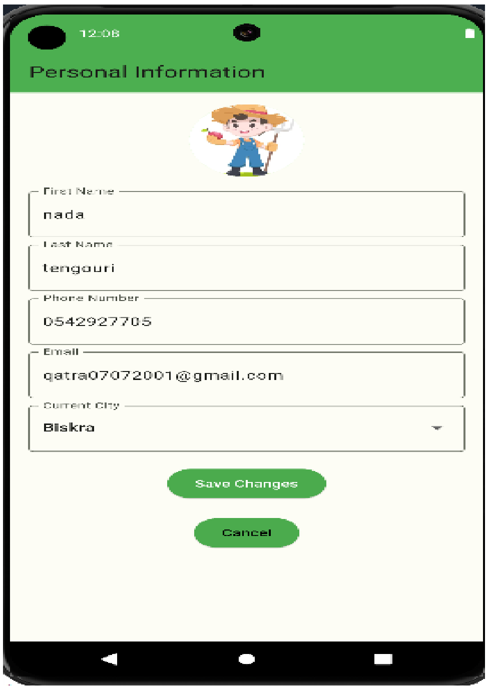


Figure 5.30: Camera activation page

7. And for the last two pages: The edit profile allows the user to edit their phone number, email address, and city. The settings page has options for account settings and notification settings. The options available include changing password, editing profile, adjusting language, and privacy & security settings. The notification settings consist of toggles for detection, elimination, and prediction.



Personal Information

First Name: nada

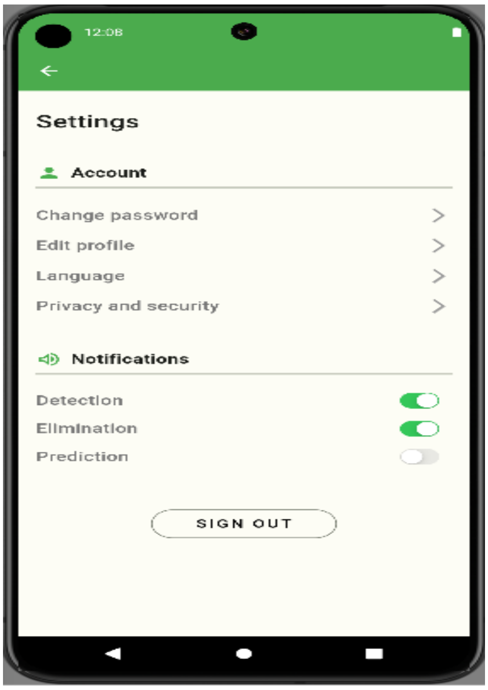
Last Name: tengouri

Phone Number: 0542927705

Email: qatra07072001@gmail.com

Current City: Biskra

Figure 5.31: Edit Profile page



Settings

Account

- Change password >
- Edit profile >
- Language >
- Privacy and security >

Notifications

- Detection ●
- Elimination ●
- Prediction ○

Figure 5.32: Settings page

5.5 Conclusion

In conclusion, this chapter has outlined the development and implementation of our system for detecting and eliminating the Tuta absoluta insect in smart greenhouses. We have detailed the programming languages, tools, libraries, and frameworks used in our project, emphasizing their crucial roles in the development process. Additionally, we have described the electronic equipment employed, including Arduino, ultrasonic sensors, aspirators, and cameras. Furthermore, we have presented the results of our work, including the performance of the CNN model (confusion matrix, accuracy, loss graphs, precision, recall, and F1-score graphs), and the functionality of the mobile application. The CNN model has demonstrated effective classification of Tuta absoluta images in real time , while the mobile application provides real-time notifications to farmers to predict if there is insect , enhancing their ability to manage Tuta absoluta insect infestations in greenhouses.

Overall, this chapter has showcased the successful development and realization of our prototype and application for farmers, which represent a significant advancement in the field of smart agriculture. Through the integration of machine learning models and electronic devices, our system offers an innovative solution for detecting and eliminating harmful insects (Tuta absoluta), ultimately improving crop yield and sustainability in agriculture.

Chapter 6

General Conclusion and Perspectives

6.1 Conclusion

In conclusion, the threat posed by Tuta absoluta to tomato cultivation in Algeria is a significant agricultural challenge. Conventional pest management strategies have often proven inadequate, leading to financial losses and environmental concerns. This research introduces an innovative solution that utilizes artificial intelligence (AI) and cutting-edge technology to detect and eradicate Tuta absoluta in smart greenhouses.

The proposed system incorporates motion sensors, temperature sensors, and cameras at greenhouse entry points to efficiently locate and eliminate tomato leaf miners. By transmitting real-time data to a deep learning algorithm for insect recognition, the system can trigger the capture and eradication process of adults, reducing reliance on ineffective traditional pest control methods.

Furthermore, the integration of a vacuum device near traps provides a precise method to attract and eliminate Tuta absoluta near greenhouse entrances, minimizing crop damage and financial losses for farmers. The accompanying mobile application delivers real-time alerts and comprehensive analytics, empowering farmers to make informed decisions and promptly protect their crops.

This innovative approach not only enhances Tuta absoluta control but also minimizes environmental impact by reducing pesticide usage. It improves crop yield and quality through effective pest control measures, addressing the challenges faced by Algerian farmers and presenting a sustainable solution for smart agriculture.

Looking ahead, future work could focus on further optimizing the device's performance, enhancing the mobile application's functionality, and exploring additional AI applications in agriculture to address other pest threats and challenges faced by farmers worldwide.

6.2 Perspectives

While our study has made significant strides in utilizing artificial intelligence for pest control in smart greenhouses, several aspects could be improved to enhance the effectiveness and applicability of our system:

1. **Enhancing AI Capabilities:** Integrating advanced **AI** algorithms to predict the movement and behavior of Tuta absoluta could improve the efficiency of the device in locating and eliminating the insect.

2. **Customization for Different Groups:** Adapting the device's design and functionality to cater to various user groups, such as farmers with different levels of expertise or specific needs, could broaden its application and usability.

3. **Enhancing Community Engagement and Awareness:** By raising awareness within the farming and agricultural community regarding the advantages of employing smart technology for pest management, it is possible to stimulate broader acceptance of these solutions, thereby promoting the implementation of more sustainable agricultural methods.

4. **Expansion to Other Pest Species:** While our focus has been on Tuta absoluta, similar approaches could be explored for other pest species that pose threats to agriculture, expanding the impact of our work in pest management.

5. **Enhancing Capabilities through Technological Integration:** By exploring synergies between our device and cutting-edge technologies like drones , we can elevate its effectiveness in pest detection and control to new heights.

6. **Developing online platform:** for receiving different client ideas, so that we will optimize our device depending on the customer opinions. This platform contain admin part that can access, add, edit and delete users information.

Ultimately, these innovations will benefit farmers and agricultural ecosystems by optimizing pest management processes.

Bibliography

[1] X. Yang, L. Shu, J. Chen, M. A. Ferrag, J. Wu, E. Nurellari, and K. Huang, “A survey on smart agriculture: Development modes, technologies, and security and privacy challenges,” *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 273–302, 2021.

[2] “Tips for getting rid of the most common pests in a greenhouse,” https://plantagreenhouses.com/blogs/learn/tips-for-getting-rid-of-the-most-common-pests-in-a-greenhouse, 2024, accessed April 29, 2024.

[3] J. James, “10+ common greenhouse pests and diseases,” https://greenhouseemporium.com/common-greenhouse-pests-diseases/, 2023, accessed April 29, 2024.

[4] D. Perdikis, E. Kapaxidi, and G. Papadoulis, “Biological control of insect and mite pests in greenhouse solanaceous crops,” *The European Journal of Plant Science and Biotechnology*, vol. 2, no. 1, pp. 125–144, 2008.

[5] E. PES, “Machine learning faqs,” https://www.einfochips.com/blog/machine-learning-faqs/, December 2019, accessed January 10, 2024.

[6] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, “Machine learning in agriculture: A review,” *Sensors*, vol. 18, no. 8, p. 2674, 2018.

[7] C. Niger, “Tuta absoluta : Introduction,” https://www.csan-niger.com/tuta-absoluta-introduction.php, January 2018, accessed February 2, 2024.

[8] M. G. Shafik, “Diagram showing the prototype of the light trap,” https://www.researchgate.net/figure/Diagram-showing-the-prototype-of-the-light-trap_fig11_264850378, 2013, accessed March 05, 2024.

[9] Campos, M. R, Biondi, Antonio, Adiga, Abhijin, Guedes, R. NC, Desneux, and Nicolas, “From the western palaearctic region to beyond tuta absoluta 10 years after invading europe,” *Journal of Pest Science*, vol. 90, pp. 787–796, 2017.

[10] M. G. Shafik, "Diagram showing the prototype of the light trap," https://www.researchgate.net/figure/Diagram-showing-the-prototype-of-the-light-trap_fig11_264850378, 2013, accessed March 05, 2024.

[11] M. S. Farooq, S. Riaz, M. A. Helou, F. S. Khan, A. Abid, and A. Alvi, "Internet of things in greenhouse agriculture: A survey on enabling technologies, applications, and protocols," *IEEE Access*, vol. 10, pp. 53 374–53 397, 2022.

[12] S. E. R. Mahr, R. A. Cloyd, D. L. Mahr, and C. S. Sadof, "Biological control of insects and other pests of greenhouse crops," *North Central Regional Publication*, vol. 581, p. 100, 2001.

[13] S. Wolfert, L. Ge, C. Verdouw, and M.-J. Bogaardt, "Big data in smart farming–a review," *Agricultural Systems*, vol. 153, pp. 69–80, 2017.

[14] A. Lytos, T. Lagkas, P. Sarigiannidis, M. Zervakis, and G. Livanos, "Towards smart farming: Systems, frameworks and exploitation of multiple sources," *Computer Networks*, vol. 172, p. 107147, 2020.

[15] B. B. Sinha and R. Dhanalakshmi, "Recent advancements and challenges of internet of things in smart agriculture: A survey," *Future Generation Computer Systems*, vol. 126, pp. 169–184, 2022.

[16] N. Gondchawar and R. Kawitkar, "Iot based smart agriculture," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, no. 6, pp. 838–842, 2016.

[17] O. Friha, M. A. Ferrag, L. Shu, L. Maglaras, and X. Wang, "Internet of things for the future of smart agriculture: A comprehensive survey of emerging technologies," *IEEE/-CAA Journal of Automatica Sinica*, vol. 8, no. 4, pp. 718–752, 2021.

[18] D. R. Harris and D. Q. Fuller, "Agriculture: definition and overview," in *Encyclopedia of Global Archaeology*. Springer New York, 2014, pp. 104–113.

[19] E. K. Gyamfi, Z. ElSayed, J. Kropczynski, M. A. Yakubu, and N. Elsayed, "Agricultural 4.0 leveraging on technological solutions: Study for smart farming sector," *arXiv preprint*, 2024.

[20] Z. Zhai, J. F. Martínez, V. Beltran, and N. L. Martínez, "Decision support systems for agriculture 4.0: Survey and challenges," *Computers and Electronics in Agriculture*, vol. 170, p. 105256, 2020.

[21] R. Patel, "The long green revolution," *The Journal of Peasant Studies*, vol. 40, no. 1, pp. 1–63, 2013.

[22] L. Lundqvist and B. Brown, “Precision agriculture: The benefits, challenges, and paths to adoption,” *Precision Agriculture*, 2024.

[23] R. Beluhova-Uzunova and D. Dunchev, “Agriculture 4.0-concepts, technologies and prospects.” *Journal of Agriculture*, 2022.

[24] S. O. Araújo, R. S. Peres, J. Barata, F. Lidon, and J. C. Ramalho, “Characterising the agriculture 4.0 landscape emerging trends, challenges and opportunities,” *Agronomy*, vol. 11, no. 4, p. 667, 2021.

[25] I. Zambon, M. Cecchini, G. Egidi, M. G. Saporito, and A. Colantoni, “Revolution 4.0: Industry vs. agriculture in a future development for smes,” *Processes*, vol. 7, no. 1, p. 36, 2019.

[26] E. Ipek, “The importance of smart agriculture practices,” *Adam Academy Journal of Social Sciences/Adam Akademi Sosyal Bilimler Dergisi*, vol. 12, no. 1, 2022.

[27] H. El Bilali and M. S. Allahyari, “Transition towards sustainability in agriculture and food systems: Role of information and communication technologies,” *Information Processing in Agriculture*, vol. 5, no. 4, pp. 456–464, 2018.

[28] G. Singh, N. Kalra, N. Yadav, A. Sharma, and M. Saini, “Smart agriculture: A review,” *Siberian Journal of Life Sciences and Agriculture*, vol. 14, no. 6, pp. 423–454, 2022.

[29] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, and K. Ueda, “Cyber-physical systems in manufacturing,” *CIRP Annals*, vol. 65, no. 2, pp. 621–641, 2016.

[30] S. Sarkar, B. Ganapathysubramanian, A. Singh, F. Fotouhi, S. Kar, K. Nagasubramanian, G. Chowdhary, S. K. Das, G. Kantor, A. Krishnamurthy *et al.*, “Cyber-agricultural systems for crop breeding and sustainable production,” *Trends in Plant Science*, 2023.

[31] Y. Ashibani and Q. H. Mahmoud, “Cyber-physical systems security: Analysis, challenges and solutions,” *Computers and Security*, vol. 68, pp. 81–97, 2017.

[32] R. Alguliyev, Y. Imamverdiyev, and L. Sukhostat, “Cyber-physical systems and their security issues,” *Computers in Industry*, vol. 100, pp. 212–223, 2018.

[33] K. M. Hosny, W. M. El-Hady, and F. M. Samy, “Technologies, protocols, and applications of internet of things in greenhouse farming: A survey of recent advances,” *Information Processing in Agriculture*, 2024.

[34] O. Postolache, J. M. Pereira, P. S. Girão, and A. A. Monteiro, “Greenhouse environment: Air and water monitoring,” *Smart Sensing Technology for Agriculture and Environmental Monitoring*, pp. 81–102, 2012.

[35] S. Kamatham, S. Munagapati, K. N. Manikanta, R. Vulchi, K. Chadipiralla, S. H. Indla, and U. S. Allam, "Recent advances in engineering crop plants for resistance to insect pests," *Egyptian Journal of Biological Pest Control*, vol. 31, pp. 1–14, 2021.

[36] J. Liu and X. Wang, "Plant diseases and pests detection based on deep learning: a review," *Plant Methods*, vol. 17, pp. 1–18, 2021.

[37] A. M. Tronsmo, D. B. Collinge, A. Djurle, L. Munk, J. Yuen, and A. Tronsmo, *Plant pathology and plant diseases*. CABI, 2020.

[38] J. Zhang, Y. Huang, R. Pu, P. Gonzalez-Moreno, L. Yuan, K. Wu, and W. Huang, "Monitoring plant diseases and pests through remote sensing technology: A review," *Computers and Electronics in Agriculture*, vol. 165, p. 104943, 2019.

[39] G. J. Messelink, J. Lambion, A. Janssen, and P. C. van Rijn, "Biodiversity in and around greenhouses: Benefits and potential risks for pest management," *Insects*, vol. 12, no. 10, p. 933, 2021.

[40] T. I. Ofuya, A. I. Okunlola, and G. N. Mbata, "A review of insect pest management in vegetable crop production in nigeria," *Insects*, vol. 14, no. 2, p. 111, 2023.

[41] L. Formisano, C. El-Nakhel, G. Corrado, S. De Pascale, and Y. Rouphael, "Biochemical, physiological, and productive response of greenhouse vegetables to suboptimal growth environment induced by insect nets," *Biology*, vol. 9, no. 12, p. 432, 2020.

[42] K. Marri, P. D. U, Shashank, and T. Sanga, *Emerging Insect Pest Threats*. Stella International Publication, 11 2023, pp. 382–404.

[43] P. G. Weintraub, E. Recht, L. L. Mondaca, A. R. Harari, B. M. Diaz, and J. Bennison, "Arthropod pest management in organic vegetable greenhouses," *Journal of Integrated Pest Management*, vol. 8, no. 1, p. 29, 2017.

[44] J. P. Yactayo-Chang, H. V. Tang, J. Mendoza, S. A. Christensen, and A. K. Block, "Plant defense chemicals against insect pests," *Agronomy*, vol. 10, no. 8, p. 1156, 2020.

[45] H. Käch, H. Mathé-Hubert, A. B. Dennis, and C. Vorburger, "Rapid evolution of symbiont-mediated resistance compromises biological control of aphids by parasitoids," *Evolutionary Applications*, vol. 11, no. 2, pp. 220–230, 2018.

[46] D. Ou, L.-M. Ren, Y. Liu, S. Ali, X.-M. Wang, M. Z. Ahmed, and B.-L. Qiu, "Compatibility and efficacy of the parasitoid eretmocerus hayati and the entomopathogenic fungus cordyceps javanica for biological control of whitefly bemisia tabaci," *Insects*, vol. 10, no. 12, p. 425, 2019.

[47] M. A. Mirhosseini, Y. Fathipour, N. Holst, M. Soufbaf, and J. Michaud, “An egg parasitoid interferes with biological control of tomato leafminer by augmentation of nesidiocoris tenuis (hemiptera: Miridae),” *Biological control*, vol. 133, pp. 34–40, 2019.

[48] M. M. Fonseca, A. Pallini, P. H. Marques, E. Lima, and A. Janssen, “Compatibility of two predator species for biological control of the two-spotted spider mite,” *Experimental and Applied Acarology*, vol. 80, pp. 409–422, 2020.

[49] C. Staff, “What is artificial intelligence? definition, uses, and types,” https://www.coursera.org/articles/what-is-artificial-intelligence, November 2023, accessed February 25, 2024.

[50] I. E. Naqa and M. J. Murphy, *Machine Learning in Radiation Oncology*. Springer, January 2015, what Is Machine Learning?

[51] I. H. Sarker, “Machine learning: Algorithms, real-world applications and research directions,” *SN Computer Science*, vol. 2, no. 3, p. 160, 2021.

[52] A. S. Gillis and M. K. Pratt, “What is machine learning and how does it work? in-depth guide,” *Machine Learning Platforms*, May 2023.

[53] J. Brownlee, “What is semi-supervised learning,” *Python Machine Learning*, December 2020.

[54] A. Shrestha and A. Mahmood, “Review of deep learning algorithms and architectures,” *IEEE Access*, vol. 7, pp. 53 040–53 065, 2019.

[55] A. Biswal, “Top 10 deep learning algorithms you should know in 2024,” https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-algorithm, February 2024, accessed March 30, 2024.

[56] I. H. Sarker, “Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions,” *SN Computer Science*, vol. 2, no. 420, August 2021.

[57] “Introduction to deep learning,” https://www.geeksforgeeks.org/introduction-deep-learning/, April 2023, accessed February 25, 2024.

[58] I. H. Sarker, “Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions,” *SN Computer Science*, vol. 2, August 2021.

[59] U. Mishra, “Best deep learning techniques,” https://www.analyticssteps.com/blogs/best-deep-learning-techniques, December 2021, accessed February 26, 2024.

[60] B. Shetty, “Classification algorithms for machine learning,” https://builtin.com/data-science/supervised-machine-learning-classification, April 2023, accessed December 30, 2023.

[61] G. Boesch, “A complete guide to image classification in 2024,” https://viso.ai/computer-vision/image-classification/, 2024, accessed January 09, 2024.

[62] S. Suman and D. K. Gupta, “Supervised image classification,” *ScienceDirect*, 2022.

[63] K. L. Kvamme and J. G. Menzer, “Unsupervised classification,” *ScienceDirect*, 2019.

[64] A. Acharya, S. Singh, and A. Dirik, “Vision transformers: A deep dive into the future of computer vision,” https://encord.com/blog/vision-transformers/, September 2023, accessed January 30, 2024.

[65] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *Computer Vision–ECCV 2020: 16th European Conference on Computer Vision, Proceedings, Part I.* Springer, 2020, pp. 213–229.

[66] T. Shehzadi, K. A. Hashmi, D. Stricker, and M. Z. Afzal, “Computer vision and pattern recognition,” *Computer Science*, June 2023, https://arxiv.org/abs/2306.04670.

[67] N. Rogge, S. Singh, and A. Dirik, “Title of the blog post,” https://huggingface.co/blog/mask2former, January 2023, accessed January 10, 2024.

[68] D. T. Illakwahhi and B. B. L. Srivastava, “Control and management of tomato leafminer-tuta absoluta (meyrick) (lepidotera, gelechiidae). a review,” *IOSR Journal of Applied Chemistry*, vol. 10, pp. 14–22, 2017.

[69] “Mineuse de la tomate,” https://www.insectosphere.fr/mineuse-de-la-tomate, accessed March 30, 2024.

[70] C. R. Ballal, A. Gupta, M. Mohan, and Y. Lalitha, “The new invasive pest *Tuta absoluta* (meyrick) (lepidoptera: Gelechiidae) in india and its natural enemies along with evaluation of trichogrammatids for its biological control,” *Current Science*, vol. 111, June 2016.

[71] A. G. S. Cuthbertson, J. J. Mathers, L. F. Blackburn, A. Korycinska, W. Luo, R. J. Jacobson, and P. Northing, “Population development of *Tuta absoluta* (meyrick) (lepidoptera: Gelechiidae) under simulated uk glasshouse conditions,” *National Library of Medicine*, vol. 4, no. 2, pp. 185–197, May 2013.

[72] S. A. Mohamed, A. G. A. Azrag, F. Obala, and S. Ndlela, “Estimating the demographic parameters of *Tuta absoluta* (lepidoptera: Gelechiidae) using temperature-dependent development models and their validation under fluctuating temperature,” *National Library of Medicine*, vol. 11, no. 2, p. 181, January 2022.

[73] T. Chen, L. Chen, J. Wang, and J. Cheng, “Development of attractants and repellents for tuta absoluta based on plant volatiles from tomato and eggplant,” *Frontiers in Sustainable Food Systems*, vol. 11, April 2023.

[74] G. U. Omonu, B. I. Omede, P. O. Ameh, and B. Bolaji, “Modelling the effects of tuta absoluta on tomato plants,” *Journal of the Nigerian Society for Mathematical Biology*, vol. 2, pp. 14–31, January 2019.

[75] N. Mujica, P. Carhuapoma, and J. Kroschel, “Tomato leafminer, *Tuta absoluta* (meyrick, 1917),” *Risk Atlas for Africa*, January 2019.

[76] EPPO, “*Tuta absoluta* (gnorab), distribution details in algeria,” https://gd.eppo.int/taxon/GNORAB/distribution/DZ, 2016, accessed March 18, 2024.

[77] J.-P. de Wit, “Tuta absoluta | how to control?” *Product specialist Crop Protection*, January 2018.

[78] A. Cocco, S. Deliperi, and G. Delrio, “Potential of mass trapping for *Tuta absoluta* management in greenhouse tomato crops using light and pheromone traps,” *Integrated Control in Protected Crops, Mediterranean Climate IOBC-WPRS Bulletin*, vol. 80, pp. 319–324, 2012.

[79] R. N. Miano, P. M. Ayelo, R. Musau, A. Hassanali, and S. A. Mohamed, “Electroantennogram and machine learning reveal a volatile blend mediating avoidance behavior by *Tuta absoluta* females to a wild tomato plant,” *Scientific Reports*, vol. 12, p. 8965, May 2022.

[80] L. Loyani and D. Machuve, “A deep learning-based mobile application for segmenting *Tuta Absoluta*’s damage on tomato plants,” *Engineering Technology and Applied Science Research*, vol. 11, no. 5, pp. 7730–7737, October 2021.

[81] K. Stewart, “Mean squared error (mse) | definition, formula, interpretation, and facts,” accessed: 2024-05-24. [Online]. Available: https://www.britannica.com/science/mean-squared-error

[82] J. FERNANDO, “R-squared: Definition, calculation formula, uses, and limitations,” https://www.investopedia.com/terms/r/r-squared.asp, May 2024, accessed June 02, 2024.

[83] T. Kanstrén, “A look at precision, recall, and f1-score,” *Towards Data Science*, Sep 2020.

[84] M. A. Hossain and M. S. A. Sajib, “Classification of images using convolutional neural networks (cnn),” *Global Journal of Computer Science and Technology*, vol. XIX, no. 13, p. 13, 2019.

[85] “Uml use case diagram tutorial,” https://www.lucidchart.com/pages/uml-use-case-diagram, 2024, accessed April 20, 2024.

[86] M. Maksimović, V. Vujović, N. Davidović, V. Milošević, and B. Perišić, “Raspberry pi as internet of things hardware: Performances and constraints,” *ResearchGate*, pp. 1–2, June 2014, conference held on June 2–5, 2014.

[87] “Raspberry pi infrared night vision camera (5mp - 222°fov),” https://store.fut-electronics.com/products/copy-of-raspberry-pi-professional-camera-night-vision-220-degree, 2024, accessed April 22, 2024.

[88] “Arduino hardware,” https://www.arduino.cc/en/hardware, April 2022, accessed April 20, 2024.

[89] A. Carullo and S. M. Marco Parvis, “An ultrasonic sensor for distance measurement in automotive applications,” *IEEE SENSORS JOURNAL*, August 2001.

[90] D. Srivastava, A. Kesarwani, and S. Dubey, “Measurement of temperature and humidity by using arduino tool and dht11,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 05, December 2018.

[91] M. S. Novelan and M. Amin, “Monitoring system for temperature and humidity measurement with dht11 sensor using nodemcu,” *International Journal of Innovative Science and Research Technology*, vol. 5, October 2020.

[92] O. Akinwole, “Design and implementation of arduino microcontroller based automatic lighting control with i^{2}c lcd display,” *Electrical & Electronic Systems*, July 2018.

[93] Python® – the language of today and tomorrow. Accessed April 14, 2024. [Online]. Available: https://pythoninstitute.org/about-python

[94] (2024, Mar) What is python? uses and applications. Accessed April 14, 2024. [Online]. Available: https://www.geeksforgeeks.org/what-is-python/

[95] C. Sabharwal, “Java, java, java,” *IEEE Potentials*, vol. 17, no. 1, pp. 33–37, Aug-Sep 1998.

[96] N. Barney, “What is c++,” *TechTarget*, August 2023.

[97] "Introduction to c++ programming language," https://www.geeksforgeeks.org/introduction-to-c-programming-language/, May 2024, accessed May 18, 2024.

[98] "Dart overview," https://dart.dev/overview, . . ., accessed juin 26, 2024.

[99] (2024, Mar) What is opencv library? Accessed April 14, 2024. [Online]. Available: https://www.geeksforgeeks.org/opencv-overview/

[100] What is tensorflow. Accessed April 14, 2024. [Online]. Available: https://www.nvidia.com/en-us/glossary/tensorflow/

[101] (2024, Feb) What is keras: The best introductory guide to keras. Accessed April 14, 2024. [Online]. Available: https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-keras

[102] (2024, Jan) Python numpy. Accessed April 14, 2024. [Online]. Available: https://www.geeksforgeeks.org/python-numpy/

[103] E. Bisong, *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. SpringerLink, Sep 2019.

[104] (2024, Jan) How to use google colab. Accessed June 02, 2024. [Online]. Available: https://www.geeksforgeeks.org/how-to-use-google-colab/

[105] "Introduction to visual studio," https://www.geeksforgeeks.org/introduction-to-visual-studio/, September 2023, accessed May 30, 2024.

[106] "Run apps on the android emulator," https://developer.android.com/studio/run/emulator, 05 2023, accessed juin 26, 2024.

[107] "Make your app the best it can be with firebase and generative ai," https://firebase.google.com/, . . ., accessed juin 26, 2024.

[108] "Arduino integrated development environment (ide) v1," https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics/, January 2014, accessed April 20, 2024.

[109] "Install raspberry pi os using raspberry pi imager," https://www.raspberrypi.com/software/, 2024, accessed April 22, 2024.