الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Master's thesis

Field: Electronics
Specialty: Embedded Systems Electronics

Presented by

LARBI AISSA Abderrahmane

&

BRAHIM Hani

Plant disease detection and treatment system using deep learning: towards an application of robotics

Proposed by: Ms. S.bouraine & Ms. Dj.naceur

ESE10

Academic Year 2024-2025

Acknowledgments

I would like to express my deepest gratitude to my supervisors, **Ms. S. Bouraine** and **Ms. D. Naceur**, for their continuous support, valuable guidance, and insightful feedback throughout this research. Their expertise and encouragement have been invaluable at every stage of this work.

I also extend my sincere thanks to the members of the examination committee for their time, constructive comments, and contributions to the improvement of this thesis.

Special thanks to **CDTA** (Centre de Développement des Technologies Avancées), **INPV** (Institut National de la Protection des Végétaux) for providing valuable information, and **ITCMI** (Institut Technique des Cultures Maraîchères Et Industrielles) for allowing us to test the system in the field. Their collaboration and the resources provided were essential for the success of this project.

Finally, I would like to thank everyone who believed in us, supported us, and stood by our side during this academic journey.

Dedications

To my beloved family, whose love and support have been my greatest strength throughout this journey.

ملخص:

نظراً لأهمية الزراعة عالمياً، وخاصةً في الجزائر حيث تُعدّ زراعة البطاطس نشاطاً زراعياً رئيسياً، لا تزال أمراض النباتات تُشكّل مشكلةً خطيرةً للمزارعين، لا سيما مرض اللفحة المتأخرة. ينتشر هذا المرض بسرعة ويُمكن أن يُسبب خسائر فادحة. في مواجهة هذا التهديد، يواصل العديد من المزارعين اللجوء إلى الممارسات التقليدية كالرش الكيميائي الأسبوعي. على الرغم من شيوع هذه الطريقة، إلا أنها غير فعّالة وقد تكون ضارة، إذ تؤدي إلى الإفراط في استخدام المواد الكيميائية، وزيادة التكاليف، والإضرار بالبيئة، وتهديد صحة المحاصيل والمزارعين.

في هذا العمل، نقترح نظاماً ذكياً لدعم القرار لإدارة أمراض المحاصيل والوقاية منها، وخاصةً مرض اللفحة المتأخرة، في محاصيل البطاطس. يدمج هذا النظام نموذج كشف بصري قائم على التعلم العميق قائم على YOLOv8 لتحديد الأمراض مبكراً، ويدمجه مع نظام تنبؤ بالطقس في الوقت الفعلي لتوقع الظروف المُواتية لحدوثها. كما يتضمن النظام محرك توصيات قادر على اقتراح وتخطيط إجراءات الرش بناءً على التحليلات التنبؤية والظروف البيئية المُثلى. يُطبّق النظام على منصة روبوتية مادية بقاعدة عجلات قابلة للتعديل ومُجهزة بآلية رش دقيقة. من خلال واجهة ويب سهلة الاستخدام، تُدمج جميع هذه الميزات عبر نظامي ROS 2 وWebSocket، مما يُمكّن المزارعين من مراقبة صحة محاصيلهم، وتلقي التنبيهات، وتقييم مستويات المخاطر، والتحكم في عمليات الرش عن بُعد.

وقد أظهرت نتائج التقييم متانة نموذجYOLOv8n ، حيث بلغ إجماليmAP@0.5 0.895 ، ودقة 0.894 للكشف عن اللفحة المتأخرة في بيانات التحقق، مما يؤكد موثوقيته في تحديد الأمراض في الحقل. ومن خلال توفير أدوات تحليل قائمة على البيانات وميزات أتمتة، يُمكّن هذا النظام المزارعين من اتخاذ قرارات أكثر استنارة، ويُقلل بشكل كبير من الرش غير الضروري. ويتمثل الهدف النهائي في تحسين الاستدامة الزراعية من خلال تقليل استخدام المواد الكيميائية، مع ضمان مكافحة فعالة للأمراض.

كلمات المفاتيح:

كشف أمراض النبات؛ الزراعة الذكية؛ التعلم العميق؛ التنبؤ بالطقس؛ نظام الرش؛ واجهة الويب؛ محاصيل البطاطس.

Résumé :

Compte tenu de l'importance de l'agriculture mondiale, et particulièrement en Algérie où la culture de la pomme de terre représente une activité agricole majeure, les maladies des plantes demeurent un problème majeur pour les agriculteurs, notamment le mildiou. Cette maladie se propage rapidement et peut entraîner des pertes dévastatrices. Face à cette menace, de nombreux agriculteurs continuent de recourir à des pratiques traditionnelles telles que la pulvérisation chimique hebdomadaire. Bien que largement utilisée, cette méthode est inefficace et potentiellement dangereuse, car elle entraîne une utilisation excessive de produits chimiques, augmente les coûts, nuit à l'environnement et met en danger la santé des cultures et des agriculteurs.

Dans ce travail, nous proposons un système intelligent d'aide à la décision pour la gestion et la prévention des maladies des cultures, notamment du mildiou, dans les cultures de pomme de terre. Ce système intègre un modèle de détection visuelle par apprentissage profond basé sur YOLOv8 pour identifier précocement les maladies, qu'il combine à un système de prévision météorologique en temps réel pour anticiper les conditions propices à leur apparition. Il intègre également un moteur de recommandation capable de proposer et de planifier des actions de pulvérisation en fonction d'analyses prédictives et de conditions environnementales optimales. Le système est déployé sur une plateforme robotisée physique à empattement réglable et équipée d'un mécanisme de pulvérisation de précision. Grâce à une interface web intuitive, toutes ces fonctionnalités sont combinées via ROS 2 et la communication WebSocket, permettant aux agriculteurs de surveiller l'état de santé de leurs cultures, de recevoir des alertes, d'évaluer les niveaux de risque et de contrôler les opérations de pulvérisation à distance.
Les résultats de l'évaluation ont démontré la robustesse du modèle YOLOv8n, avec un mAP@0,5 global de 0,895 et une précision de 0,894 pour la détection du mildiou sur les données de validation, confirmant sa fiabilité pour l'identification des maladies sur le terrain. Grâce à des outils d'analyse basés sur les données et à des fonctions d'automatisation, ce système permet aux agriculteurs de prendre des décisions plus éclairées et de réduire considérablement les pulvérisations inutiles. L'objectif ultime est d'améliorer la durabilité agricole en minimisant l'utilisation de produits chimiques tout en garantissant une lutte efficace contre les maladies.

Mots clés : Détection des maladies des plantes ; Agriculture intelligente ; Apprentissage profond ; Prévisions météorologiques ; Système de pulvérisation ; Interface web ; Cultures de pomme de terre ;

Abstract: Given the importance of agriculture worldwide, and particularly in Algeria where potato cultivation represents a major agricultural activity, plant diseases remain a serious problem for farmers, particularly late blight. This disease spreads rapidly and can cause devastating losses. Faced with this threat, many farmers continue to resort to traditional practices such as weekly chemical spraying. Although widely used, this method is ineffective and potentially harmful, as it leads to excessive use of chemicals, increases costs, harms the environment, and endangers both crop and farmer health.

In this work, we propose an intelligent decision support system for managing and preventing crop diseases, particularly late blight, in potato crops. This system integrates a deep learning visual detection model based on YOLOv8 to identify diseases early, which it combines with a real-time weather forecasting system to anticipate conditions conducive to their occurrence. It also incorporates a recommendation engine capable of proposing and planning spraying actions based on predictive analyses and optimal environmental conditions. The system is deployed on a physical robotic platform with an adjustable wheelbase and equipped with a precision spraying mechanism. Through an intuitive web interface, all these features are combined via ROS 2 and WebSocket communication, allowing farmers to monitor the health of their crops, receive alerts, assess risk levels, and control spraying operations remotely. The evaluation results demonstrated the robustness of the YOLOv8n model, with an overall mAP@0.5 of 0.895 and an accuracy of 0.894 for late blight detection on the validation data, confirming its reliability for identifying diseases in the field. By providing data-driven analysis tools and automation features, this system allows farmers to make more informed decisions and significantly reduce unnecessary spraying. The ultimate goal is to improve agricultural sustainability by minimizing chemical use while ensuring effective disease control.

Keywords: Plant disease detection; Smart agriculture; Deep learning; Weather forecasting; Spraying system; Web interface; Potato crops.

List of acronyms and abbreviations

AI: Artificial Intelligence

AP: Average Precision

BABA: ß-aminobutyric acid

CAM: Camera

CLI: Command Line Interface

CNNs: Convolutional Neural Networks

DAT: Days After Transplanting

DL: Deep Learning

DRL: Deep Reinforcement Learning

DSS: Decision Support Systems

DWD: Deutscher Wetterdienst

ECMWF: European Centre for Medium-Range Weather Forecasts

FN: False Negatives

FP: False Positives

GDP: Gross Domestic Product

GFS: Global Forecast System

GPU: Graphics Processing Unit

INPV : Institut National de la Protection des Végétaux

IoT: Internet of Things

IoU: Intersection over Union

ITK: Indigenous Technical Knowledge

JSON: JavaScript Object Notation

mAP: mean Average Precision

ML: Machine Learning

NMITCON: Networks, Multimedia and Information Technology

PDI: Percent Disease Index

PID: Process ID

PLRV: Potato Leafroll Virus

RF: Random Forest

ROS: Robot Operating System

SGD: Stochastic Gradient Descent

SVM: Support Vector Machines

TCP: Transmission Control Protocol

TP: True Positives

YOLO: You Only Look Once

Table of Contents

List of figures

List of tables

General introduction

For millennia, agriculture has served as the primary means of human sustenance. This pivotal shift from nomadic hunting and gathering to settled cultivation of crops and domestication of animals enabled the establishment of permanent communities and fostered societal development.

Today, agriculture's importance persists. It not only provides the global food supply but also underpins the livelihoods of vast populations, particularly in rural regions. With a continually growing world population, agriculture plays an increasingly critical role in ensuring global food security and promoting sustainable environmental stewardship.

Agriculture remains a cornerstone of Algeria’s economy, contributing 12.9% to the national Gross Domestic Product (GDP) in 2023 and ranking as the third-largest sector [1].

Algerian farmers currently face numerous challenges in maintaining crop health and achieving optimal yields. Among the most significant threats are plant diseases, which can severely damage crops, reduce food production, and result in substantial financial losses. In severe instances, these diseases can lead to the devastation of entire agricultural fields. Without timely detection and appropriate intervention, plant diseases can proliferate rapidly, causing long-term detriment to agricultural productivity and food security.

Despite its economic significance and the considerable threats to agricultural output, the sector in Algeria continues to rely heavily on traditional practices. Farmers frequently employ rudimentary tools and manual techniques, leading to inefficiencies in crop monitoring, delays in disease detection, and the improper or excessive use of pesticides.

These shortcomings contribute to diminished crop quality, financial losses, and environmental degradation. Human-dependent methodologies are inherently constrained by inconsistent monitoring, diagnostic inaccuracies, and labor-intensive processes, limitations that are particularly acute in large-scale farming operations. For example, manual pesticide application not only poses risks to human health but also exacerbates ecological damage through over-application.

Problem Statement:

There is a pressing need in Algeria for an intelligent system capable of early and real-time detection and management of destructive plant diseases. Current traditional methods for disease identification are often slow and labor-intensive, leading to delayed interventions, significant crop losses, and the excessive application of agrochemicals. An automated, intelligent solution would substantially improve disease control efficacy, reduce economic losses, and promote more sustainable agricultural practices.

Our main objectives are:

a. To **minimize** chemical product usage by enabling targeted treatment application only where necessary.

b. To **maximize** crop preservation through early and precise intervention against diseases.

c. To **reduce** human health risks by limiting direct exposure to harmful agrochemicals.

d. To **enhance** operational efficiency by automating disease detection and treatment processes, thereby saving labor and time.

e. To **empower** farmers with detailed data on crop health and environmental conditions, facilitating more informed decision-making for the effective management of threats such as late blight.

Solution:

To address the challenges posed by late blight in potato crops, we propose an **integrated intelligent system** for efficient disease management. This system incorporates several key technologies: a **deep learning-based AI model** for real-time detection of late blight symptoms on potato leaves; **weather forecasting capabilities** for predictive early risk assessment; and an **automated spray mechanism** to facilitate precision agriculture techniques [2, 3]. Furthermore, a **web-based interface** will provide continuous data streaming, enabling farmers to monitor crop health, receive timely alerts, and make data-driven decisions. This comprehensive solution aims to significantly reduce crop losses, minimize reliance on chemical treatments, and support sustainable agricultural practices in Algeria.

General Methodology and System Architecture:

The proposed system is directed toward the real-time detection of late blight, in addition to the prediction of potential infections. Its objective is to provide precise timing for spray actions to mitigate disease spread. The entire system will be integrated into a robotic platform and managed through a user-friendly interface to facilitate ease of use for the farmer.

The Main System: The core of the system is based on an AI model that performs real-time detection of visual disease symptoms. This detection module communicates directly with a central decision-making logic. Concurrently, a weather-based prediction algorithm analyzes meteorological data fetched from a weather API to forecast the risk of potential infections. Information from both the AI detection and the weather prediction modules is fed into the decision-making logic, which then determines if a spray command should be issued and precisely when the intervention is most optimal.

System Integration: To translate this system from concept to reality, it must be integrated into a physical, mechanical robotic platform, enabling it to operate within a real agricultural environment. This integration includes the installation of an action mechanism, specifically the precision sprayer, which ensures that treatment is applied

effectively with minimal losses. To simplify the operation for the end-user, a web-based user interface will be integrated, allowing the farmer to command, monitor, and manage the entire system remotely.

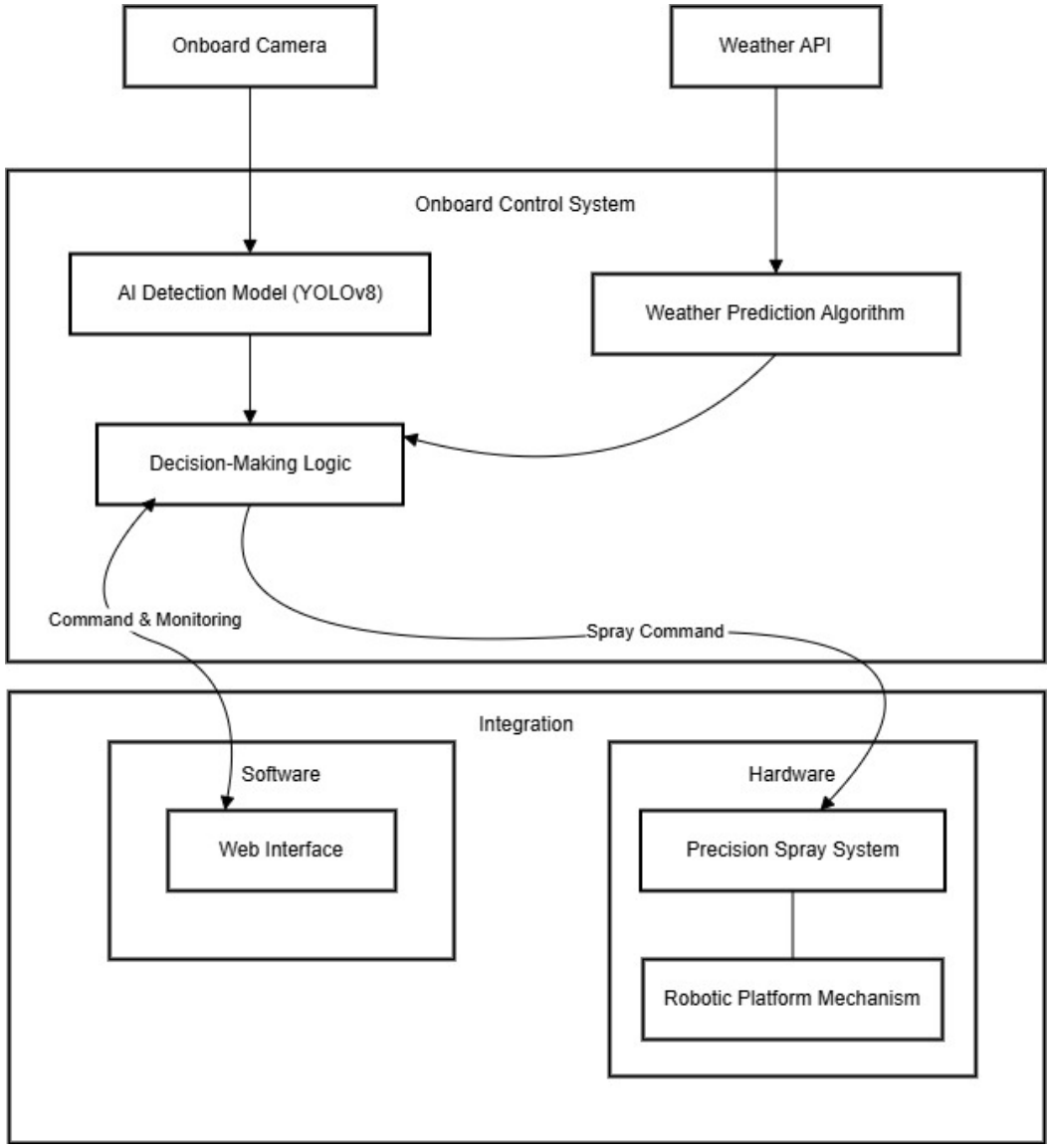


Figure 0.1: *general methodology and system architecture*

Chapter 1 State of the Art

1.1 Overview of Traditional Plant Disease Detection Techniques Treatment Methods

Traditional plant disease detection has historically relied on farmers' direct visual inspection for symptoms such as leaf spots, discoloration, wilting, or characteristic lesions like the "burning of leaves" in potato late blight [4]. This visual assessment is often guided by ancestral knowledge and Indigenous Technical Knowledge (ITK), reflecting generations of experiential learning within specific agroecological contexts [5]. Upon identification of an infection, management strategies primarily involved a suite of cultural control methods aimed at prevention and ecological balance. These included practices such as sanitation (e.g., removal of infected plant debris), crop rotation, use of disease-free planting material, and careful soil and site management [6]. Additionally, farmers employed local remedies, frequently ethnobotanical in origin and developed through empirical observation, which included plant-derived concoctions, ash, and animal-based products [5]. While these integrated traditional systems are foundational and demonstrate a profound understanding of local ecosystems, their efficacy can be constrained by the subjectivity inherent in visual diagnosis, variability in the effectiveness of local treatments, and considerable labor intensity [7], thereby highlighting areas where advanced technologies may offer improvements.

1.2 Overview of Artificial Intelligence Techniques in Agriculture

The field of Artificial Intelligence (AI) encompasses a hierarchy of techniques, from foundational Machine Learning to advanced hybrid models, each with specific applications in modern agriculture (see Figure 1.1).

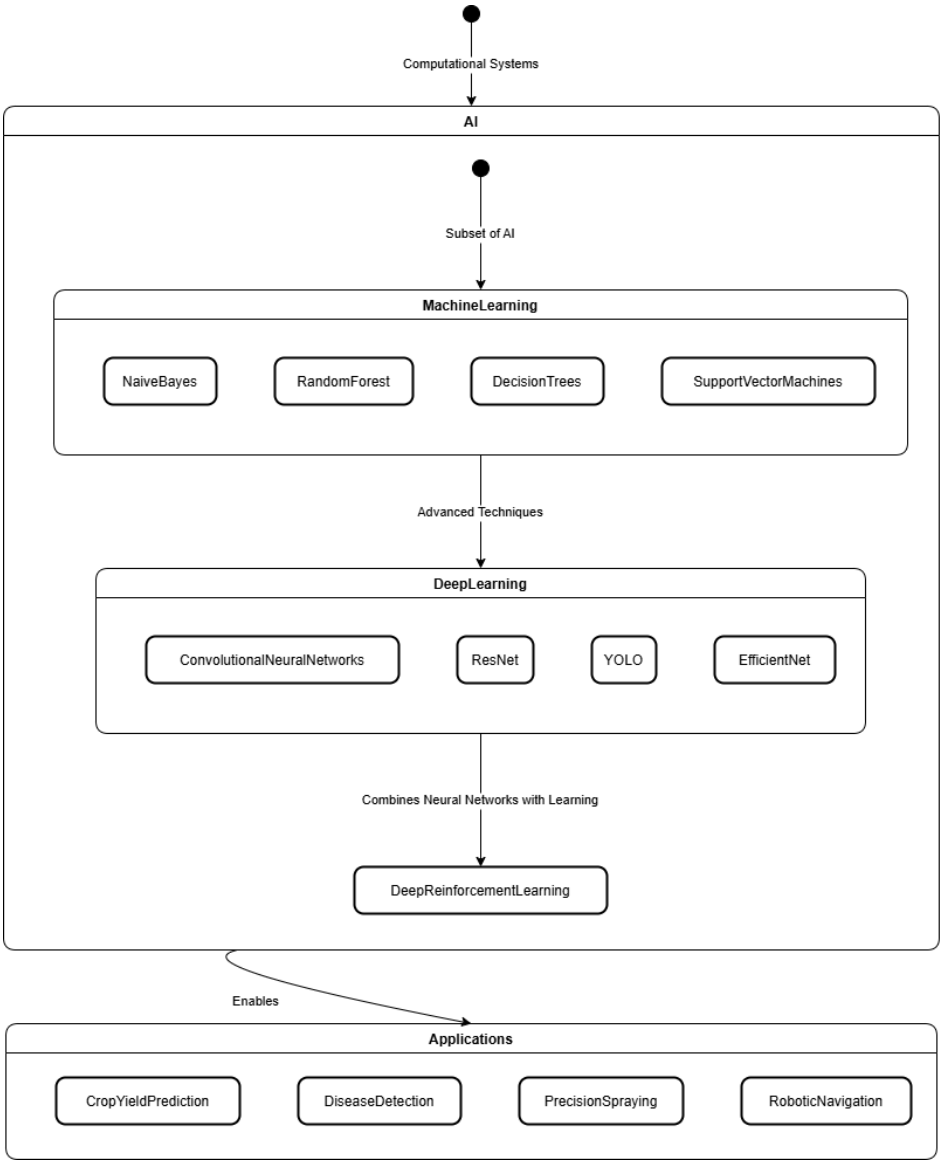


Figure 1.1: *A flowchart illustrating the hierarchy of Artificial Intelligence techniques and their applications in agriculture.*

1.2.1 Artificial Intelligence (AI)

Artificial Intelligence (AI) refers to computational systems capable of performing tasks that traditionally require human intelligence, such as problem-solving and decision-making [8]. In agriculture, AI has emerged as a transformative tool, enabling applications such as crop yield prediction, disease detection, and precision pesticide spraying. By automating resource-intensive processes, AI can reduce operational costs, minimize human error, and enhance scalability. For example, AI-driven agricultural robots now perform tasks ranging from soil analysis and seed planting to irrigation optimization, leveraging real-time data on soil moisture and nutrient levels [9]. AI also plays a critical role in pest management, where automated systems conduct regular crop inspections to identify early signs of disease or pest damage, thereby enabling timely interventions [9].

1.2.2 Machine Learning (ML)

Machine Learning (ML), a subset of Artificial Intelligence (AI), encompasses algorithms designed to analyze data and generate predictions through statistical analysis [8]. While ML has long been a cornerstone of AI applications, advancements in computational power and reduced storage costs in recent years have catalyzed the emergence of more sophisticated techniques, such as deep learning [10]. Traditional ML algorithms remain widely used for solving smaller-scale classification problems, including text analysis. For instance, Naive Bayes [11, 12] operates under the assumption of independence between input features, making it a simple yet effective choice for classification tasks. Random Forest [13], on the other hand, aggregates multiple decision trees to model intricate relationships within datasets, while standalone Decision Trees [14] structure data hierarchically to support rule-based decision-making. Support Vector Machines (SVM) [15] classify data by constructing hyperplanes that optimally separate features into distinct categories [10]. Stochastic Gradient Descent (SGD) [16], in comparison, optimizes model parameters through iterative updates on data subsets, enabling efficient training for large-scale linear models like logistic regression. Despite their utility, these conventional algorithms are primarily suited for low-complexity tasks and often struggle

with scalability when confronted with large, heterogeneous, or high-dimensional inputs, such as high-resolution agricultural imagery or real-time sensor data.

1.2.3 Deep Learning (DL)

Deep Learning (DL), an advanced branch of ML, utilizes multi-layered neural networks to autonomously extract hierarchical features from raw data [8]. The advent of DL has revolutionized image-based plant disease detection, with Convolutional Neural Networks (CNNs) [17] achieving state-of-the-art performance. Architectures like ResNet [18], which addresses vanishing gradient issues through residual connections, and YOLO (You Only Look Once) [19], a single-stage object detection framework, are particularly effective in identifying diseases from leaf images [20]. EfficientNet [21] further optimizes model scalability by balancing network depth, width, and resolution, which enables efficient deployment on resource-constrained devices [20]. These models excel at early disease diagnosis and can even predict susceptibility to future outbreaks when trained on robust, well-curated datasets.

1.2.4 Deep Reinforcement Learning (DRL)

Deep Reinforcement Learning (DRL) combines deep neural networks with reinforcement learning, enabling systems to learn optimal decision-making policies through trial-and-error interactions with their environment [22]. In agriculture, DRL has shown promise in applications such as robotic navigation and precision spraying. For example, DRL-trained robots can autonomously map and navigate unstructured farm environments, avoiding obstacles while optimizing pesticide application routes [23]. Unlike traditional vision systems that require extensive labeled data, DRL simplifies tasks like visual navigation by dynamically adapting to feedback, making it suitable for real-time, unstructured scenarios [24].

1.2.5 Hybrid Models

Recent advancements integrate DL for visual perception with DRL for robotic control, creating hybrid systems that combine accuracy and adaptability. For instance, CNNs can detect diseased plant regions, while DRL algorithms guide the robot's spraying mechanism to target affected areas precisely. This approach minimizes chemical usage, reduces environmental impact, and ensures scalable deployment across diverse agricultural settings.

1.3 Comparative Analysis of AI Techniques for Plant Disease Detection

The development of an effective Plant Disease Detection and Treatment System necessitates a careful selection of the underlying AI architecture. Machine Learning (ML), Deep Learning (DL), and Deep Reinforcement Learning (DRL) present distinct profiles in terms of accuracy, computational efficiency, and data requirements, as summarized in (Table 1.1). These differences significantly impact their suitability for real-world agricultural applications, particularly for deployment on robotic platforms. Consequently, an evaluation across these critical criteria is essential.

1.3.1 Accuracy and Model Performance

Regarding accuracy in plant disease detection, which determines the precision of classification, different AI paradigms offer varied performance. Traditional ML models, such as Support Vector Machines (SVM), Random Forest (RF), and Stochastic Gradient Descent (SGD), provide structured classification. For instance, SVM has demonstrated capabilities like achieving 87% accuracy in citrus disease detection [25]. However, a primary limitation of ML is its reliance on manual feature extraction, a process that often struggles to capture the complex patterns indicative of plant diseases; RF, for example, achieved only 76.8% accuracy in one study [25]. In contrast, Deep Learning, particularly

Convolutional Neural Networks (CNNs) like VGG-16 [26] and Inception-v3 [27], excels by automatically extracting hierarchical features from image data. This capability leads to superior accuracy, with models like VGG-16 achieving 89.5% [25, 28, 29]. Despite their strengths, DL models demand large datasets and face risks of overfitting when data is small or imbalanced [30, 31]. Deep Reinforcement Learning, while offering dynamic adaptation to environments and potential for optimizing real-time robotic decision-making, is less directly suited for primary disease detection tasks due to its computational intensity and the complexity of designing effective reward structures for classification. In summary, for disease classification accuracy, DL models, especially CNNs, generally provide the highest performance due to their advanced feature learning. ML techniques can be suitable for smaller datasets where manual feature engineering is feasible, whereas DRL's strengths currently lie more in autonomous navigation and control rather than direct disease classification.

1.3.2 Computational Efficiency for Robotic Deployment

Computational efficiency for real-time robotics is another crucial factor, as deployment often involves embedded systems like Raspberry Pi, requiring a balance between accuracy and processing speed. Machine Learning models typically exhibit low computational overhead, making them feasible for resource-limited systems [32]. However, their inherently lower accuracy compared to DL can limit their effectiveness in complex, variable agricultural environments [33]. Deep Learning models, while offering high accuracy, generally require GPUs for training and can be resource-intensive for inference. Nevertheless, optimization techniques such as model pruning and quantization are increasingly enabling the deployment of DL models on edge devices, albeit with careful consideration [34]. Unoptimized DL models can significantly hinder real-time performance [35, 36]. Deep Reinforcement Learning is adaptable to dynamic robotics tasks but also presents high computational demands and complex reward design challenges that can impede real-time viability [32]. Therefore, ML often stands out for its computational efficiency on constrained systems. DL requires specific hardware acceleration or significant optimization for real-time field deployment, and DRL’s real-

time feasibility often depends on the complexity of the learned policy and the efficiency of its post-training deployment.

1.3.3 Data Requirements and Scalability

Data requirements and scalability are critical considerations, as the quality and volume of data directly influence model performance and its ability to generalize. Traditional ML approaches require manual feature engineering and can function with smaller datasets, but they often struggle with high-dimensional inputs like raw imagery [37, 35]. Deep Learning models, conversely, demand large, meticulously annotated datasets; architectures like ResNet and EfficientNet, for example, typically require thousands of labeled images to achieve robust performance [38–40]. While this is a significant hurdle, techniques such as transfer learning can mitigate data scarcity by leveraging knowledge from pre-trained models, adapting them to specific agricultural tasks with smaller, domain-specific datasets [41, 42]. Deep Reinforcement Learning learns through direct interaction with its environment, bypassing the need for large pre-labeled datasets in the traditional sense, but this learning process itself requires extensive computational resources and careful reward tuning. Key insights reveal that DL's high accuracy is intrinsically linked to the availability of large, high-quality datasets, which are often costly and time-consuming to collect and annotate [39, 43]. While transfer learning and few-shot learning strategies are being developed to reduce these data dependencies, they still necessitate careful domain adaptation [41, 42]. Furthermore, real-world variability in agricultural settings, such as inconsistent lighting, diverse leaf orientations, and varying stages of disease, poses significant challenges to dataset generalization for all image-based approaches [40, 43]. In essence, both ML and DL rely heavily on labeled data, with DL having the most substantial requirements. DRL offers an alternative learning paradigm that avoids direct data annotation for classification but is generally considered impractical for standalone disease classification when compared to supervised learning techniques optimized for this task.

Table 1.1: *Comparative Analysis Table*

Criteria	**Machine Learning (ML)**	**Deep Learning (DL)**	**Deep Reinforcement Learning (DRL)**
Accuracy	- **Moderate accuracy** (e.g., SVM: 87%)	- **Highest accuracy** (e.g., VGG-16: 89.5%)	- **Not directly tested**, but potential for adaptive decision-making
	- Limited by manual feature extraction	- Automatic feature extraction improves detection	- Requires complex reward design for disease tasks
Computational Efficiency	- **Low computational demand**, suitable for embedded systems	- **High demand** (GPUs required) but optimizable via pruning/quantization	- **Extremely resource-intensive**, challenges for real-time deployment
	- Real-time feasible but less accurate in dynamic environments	- Edge deployment possible with optimization (e.g., TensorFlow Lite)	- Suited for post-training navigation tasks
Data Requirements	- **Small datasets** with manual feature engineering	- **Large labeled datasets** required	- **Learns via interaction**, minimal direct annotation
	- Struggles with high-dimensional data	- Transfer learning mitigates data scarcity	- Requires extensive trial-and-error for reward tuning

Conclusion:

DL (CNNs) is optimal for high-accuracy disease detection but requires dataset and computational optimization. ML suits low-resource environments, while DRL's niche lies in robotic navigation and adaptive control. These key trade-offs are summarized in (Table 1.2). Ultimately, hybrid architectures (e.g., CNN + DRL) may balance vision and action for precision agriculture.

Table 1.2: *Summary of Techniques*

Technique	Best For	Limitations
ML	Small datasets, low-resource environments	Low accuracy, manual feature engineering
DL	High-accuracy detection, hierarchical features	Data-hungry, computationally intensive
DRL	Adaptive robotics navigation and control	Complex reward design, high training costs

1.4 Literature Review of Deep Learning Models

Recent advancements in plant disease detection have predominantly leveraged Deep Convolutional Neural Networks (CNNs) with transfer learning, while Machine Learning (ML) and Deep Reinforcement Learning (DRL) remain niche due to limitations in scalability and adaptability. Below, we focus on DL approaches, particularly YOLO-based architectures, to justify the selection of YOLOv8 for our project.

1.4.1 Foundational CNNs for Image Classification

Early and influential work with traditional CNNs demonstrated their potential in agricultural image analysis. For example, Mohanty et al. (2016) [44] utilized AlexNet [45] and GoogLeNet [46], training them on a substantial dataset of 54,306 images from PlantVillage [47]. Their results were compelling: GoogLeNet, when employing transfer learning, achieved an impressive 99.35% accuracy in classifying 26 diseases across 14 different crop types. In contrast, AlexNet, when trained from scratch, reached an accuracy of 85.53%. Similarly, Liu et al. [48] applied an AlexNet model to a dataset of 13,689 apple leaf images, successfully achieving 97.62% accuracy in detecting various apple diseases. These traditional CNN approaches offer significant advantages, primarily through their automated feature extraction capabilities, which eliminate the need for laborious manual preprocessing. Furthermore, the application of transfer learning has proven effective in mitigating issues related to data scarcity. However, these models also present notable limitations, including high computational costs, often necessitating GPU-dependent infrastructure, and restricted real-time feasibility when considered for deployment on field robotics.

1.4.2 Real-Time Object Detection with YOLO

More recently, a critical distinction emerged between image classification and object detection. While models like AlexNet and GoogLeNet are powerful for classification (telling an operator if a disease is present in an image), object detection models like YOLO (You Only Look Once) can identify if a disease is present and, crucially, where it

is located within the image. This capability is far more useful for a robotic spraying system, which needs precise coordinates for targeted treatment. YOLO-based architectures have therefore gained prominence, particularly because they excel in real-time performance. For instance, a hybrid approach using YOLOv7 combined with a CNN classifier [49] achieved a remarkable 98.8% accuracy in tomato leaf disease detection and classification, effectively merging YOLO's processing speed with the classification precision of CNNs. Another widely recognized model, YOLOv5 [50], demonstrated 93% accuracy in the real-time identification of tomato diseases, showcasing a strong balance between speed and accuracy suitable for field deployment. The primary advantages of YOLO-based architectures are their capacity for real-time processing and their single-stage detection mechanism. This allows them to efficiently localize and classify diseases in a single pass, making them highly suitable for applications demanding rapid response and operational efficiency.

To provide a clear side-by-side comparison, the performance metrics and key characteristics of these influential DL models are summarized in (Table 1.3). This analysis highlights the trade-offs between accuracy, speed, and deployment feasibility, setting the stage for the selection of an optimal architecture.

Table 1.3: *Comparative Analysis of DL Models*

Model	Accuracy	Speed	Key Strength	Limitation
AlexNet	85.53–97.62%	Low	Foundational CNN architecture	Computationally intensive
GoogLeNet	99.35%	Moderate	High accuracy with transfer learning	Requires large datasets
YOLOv5	93%	High	Real-time field deployment	Lower accuracy than CNNs
YOLOv7	98.8%	High	Speed-accuracy balance	Complex implementation
YOLOv8 (Proposed)	*99%+ (Expected)*	**Very High**	Optimized architecture for precision	Requires fine-tuning for agriculture

1.4.3 Why YOLOv8?

YOLOv8, as the latest most stable iteration in the YOLO family, builds upon the established successes of its predecessors like YOLOv5 and YOLOv7, offering several key enhancements that make it particularly suitable for our project. Firstly, it provides enhanced accuracy resulting from improvements in its backbone and neck architectures, which lead to better feature extraction capabilities. Secondly, YOLOv8 is engineered for faster inference speeds and is specifically optimized for edge devices, such as the

NVIDIA Jetson platform, a critical consideration for deployment on mobile agricultural robots. Furthermore, its adaptability is improved through advanced data augmentation techniques, enabling it to perform more robustly in dynamic agricultural environments characterized by variable lighting conditions and potential occlusions. Finally, YOLOv8 demonstrates strong scalability and compatibility with hybrid systems, for instance, facilitating integration with Deep Reinforcement Learning (DRL) for tasks like advanced robotic navigation and control, thereby supporting comprehensive precision agriculture solutions.

1.5 Dataset Selection for Vision System

The performance of a computer vision system critically hinges on the quality, diversity, and relevance of the dataset used for its training and validation. A well-curated dataset is paramount as it ensures robust model generalization, enables reliable deployment in real-world agricultural scenarios, and aligns with the operational constraints inherent in robotic vision systems.

1.5.1 Selection Criteria

To select an optimal dataset for plant disease detection, we prioritize several key criteria: diversity, encompassing variability in plant species, disease types, and imaging conditions (such as lighting and angles); size, ensuring an adequate volume of images to prevent overfitting and promote robust learning; quality, indicated by high-resolution images with clear and accurate annotations; and compatibility with robotic vision, which demands real-world applicability, including diverse backgrounds and authentic field conditions.

1.5.2 Critical Analysis

A side-by-side comparison of prominent public datasets against these criteria is presented in (Table 1.4). On the positive side, several large datasets offer ample data for training sophisticated deep learning models. For example, PlantifyDR [51] contains approximately 125,000 images, and the widely-used PlantVillage dataset [47] includes around 54,000 images. In terms of class diversity, datasets like New-Plant-Diseases and PlantVillage are notable for covering up to 38 distinct classes, thereby enabling the development of multi-disease detection systems.

However, significant limitations also exist. A common issue is the presence of unified backgrounds; many prominent datasets, including PlantVillage and Tomato [52], primarily feature images taken in controlled laboratory environments. This lack of background variability can limit a model's ability to generalize to real-world field conditions. Conversely, some available datasets are relatively small. For instance, Plant-Doc [53], with about 4,500 images, risks model underfitting if used as a standalone resource for training complex models.

These observations highlight key trade-offs in dataset selection. There is often a tension between dataset volume and generalization. While larger datasets like PlantifyDR [51] provide extensive training material, they might risk overfitting if the model is intended for resource-constrained robotic platforms, or if the diversity within the large dataset doesn't match the target environment. Smaller datasets such as Plant-Doc [53], on the other hand, might lack the sheer volume for optimal scalability of deep learning models. Another crucial trade-off involves field relevance. Datasets like Plant-Doc [53], despite their smaller size, often feature more varied imaging conditions (e.g., diverse lighting, natural backgrounds) that better mirror real-world robotic operations. However, to be effectively used, these datasets typically require significant data augmentation to artificially increase their size and variability for training purposes.

Table 1.4: *Dataset Comparison*

Dataset	Data Size	Diversity	Quality	Robotic Vision Compatibility
New-Plant-Diseases[54]	87,000	38 classes (plants + diseases)	256×256, JPG	Unified background
Tomatoleaf[55]	10,000	10 tomato diseases	256×256, JPG	Unified background
Tomato[52]	18,000	10 tomato diseases	256×256, JPG	Unified background
PlantVillage[47]	54,000	38 classes (14 plants + 26 diseases)	256×256, JPG	Unified background
PlantifyDR-Dataset[51]	125,000	10 plants, 37 diseases	256×256, JPG	Unified background
Plant-Doc-Dataset[53]	4,500	30 classes (13 plants)	900×675, JPG	Varied field conditions

1.5.3 Final Selection

To effectively balance data volume, diversity, and crucial field applicability, we propose a hybrid dataset strategy. This approach combines the strengths of two distinct datasets: PlantVillage [47] will serve as the primary source for training data, leveraging its substantial volume of approximately 54,000 images and broad coverage of 38 classes. Its main advantage lies in providing a balanced class distribution and comprehensive disease representation, which is essential for initial model learning. Complementing this, the Plant-Doc [53] dataset is designated for validation and fine-tuning purposes. Although smaller, with around 4,500 images across 30 classes, its key advantage is its

composition of images captured under real-world field conditions. This two-stage approach is strategic: the variety of disease morphologies in PlantVillage allows the model to build a foundational feature set for identification, while the challenging conditions and potential confounders (e.g., environmental artifacts that mimic symptoms) in Plant-Doc are critical for enhancing the model's accuracy and robustness for operational deployment.

1.6 The Robot Operating System (ROS) Framework

The Robot Operating System (ROS) is a pivotal open-source middleware framework engineered to facilitate the development of complex robotic systems. Despite its name, ROS is not a conventional operating system. Instead, it functions as a flexible meta-operating system, providing an abstraction layer that operates on top of a host OS such as Ubuntu. ROS furnishes a comprehensive suite of tools, libraries, and standardized conventions that streamline the creation of modular, distributed, and scalable robot software [56, 57]. The widespread adoption of ROS in contemporary robotics underscores its capability to provide a standardized and scalable environment for building intelligent systems. Its impact is evident across a multitude of domains, including autonomous vehicles, drones, medical robotics, and, increasingly, agricultural automation [56].

1.6.1 ROS Applications in the Agricultural Sector

Several projects and academic works have successfully integrated ROS to address key challenges in agriculture. These examples demonstrate the framework's practical utility:

- **Irrigation Optimization:** One project focused on developing an autonomous robotic system to improve irrigation management, using ROS 2 to create a simulated agricultural environment for development and testing before deployment on a physical robot [58].

- **Autonomous Navigation:** A notable thesis addressed the challenge of outdoor robot navigation by integrating a full sensor suite (LiDAR, GPS, cameras) with the ROS 2 Nav2 stack. It used a custom computer vision pipeline with a YOLOv8 neural network to allow a robot to better discern traversable terrain, such as tall grass, from impassable obstacles [59].
- **Digital Twin Emulation:** The "AgROS" project developed a ROS-based emulation tool to bridge the gap between software-based decision support systems and physical machinery. The tool allows for the predictive analysis and testing of modules that can be directly transferred to real-world robots, aiming to create a "digital twin" for optimizing agricultural operations [60].

1.6.2 The Modular Architecture of ROS

The power of ROS lies in its ability to decompose complex robotic applications into smaller, independent, and reusable components called **nodes**. These nodes communicate with each other using a standardized messaging system, which enhances scalability, code reuse, and ease of debugging.

a. *Nodes: The Building Blocks of a ROS System*

In ROS, a node is the smallest executable unit, typically a process responsible for a single, well-defined task. For example, a simple mobile robot might have separate nodes for:

- Capturing and publishing camera data.
- Processing sensor data to detect obstacles.
- Controlling the wheel motors.

Each node is a standalone program, which allows them to be developed, tested, and modified independently.

b. *Topics: The Communication Buses*

Nodes communicate using a publish-subscribe model facilitated by named buses called **topics**. A node with data to share (e.g., a camera node) **publishes** messages to a topic. Any other node that needs this data (e.g., an object detector or a display node) can **subscribe** to that topic to receive the messages.

This system decouples the nodes from one another; the publishing node does not know or care which nodes are subscribing. This creates a highly flexible and scalable architecture. The communication itself is handled by the underlying ROS middleware. In the modern ROS 2, this is the Data Distribution Service (DDS), a robust protocol that enables real-time, secure, and peer-to-peer data transfer with automatic discovery of nodes and topics.

The specific architecture for a given application can be visualized in a communication graph. (**Figure 1.2)** illustrates a conceptual graph for a hypothetical agricultural robot. In this example, sensor nodes like “blight_detector” and “weather_reporter” publish their data to dedicated topics. A central “spray_scheduler” node subscribes to this information to make decisions, demonstrating how the modular design allows for a clear and logical flow of data through the system.

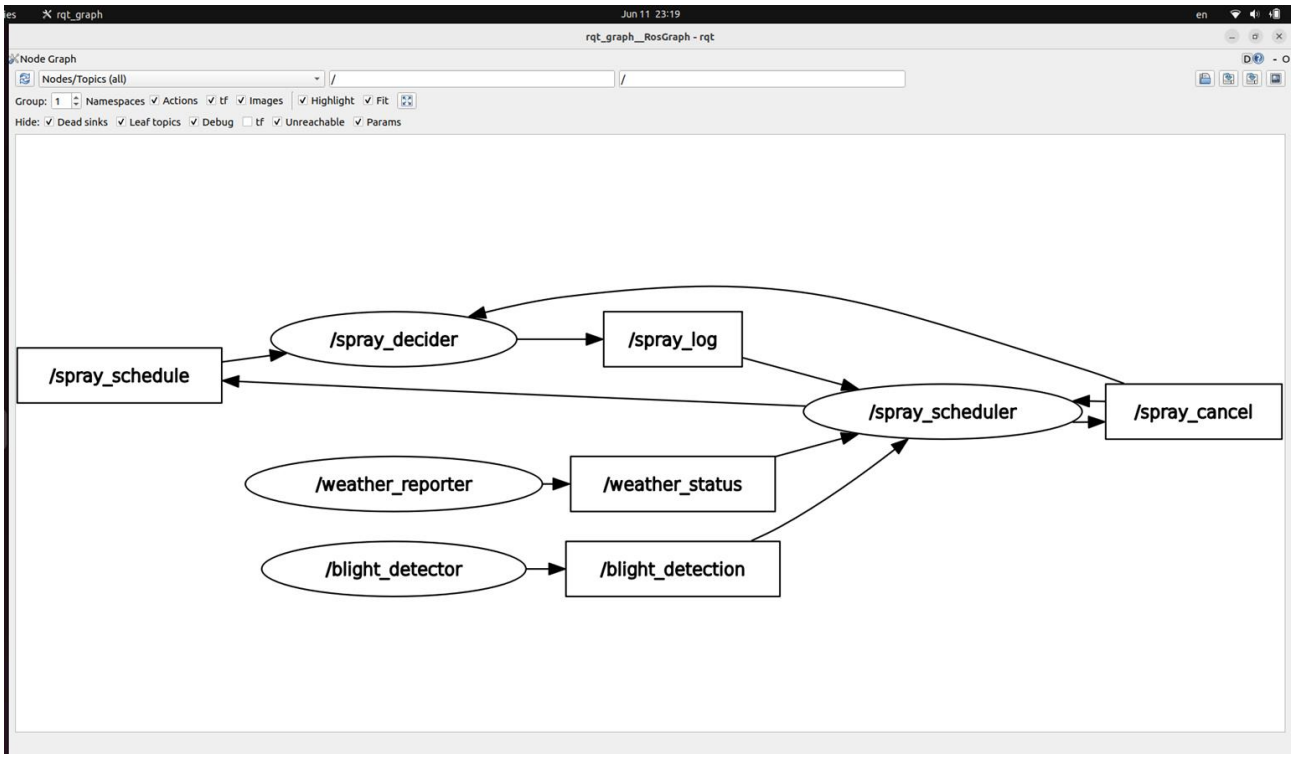


Figure 1.2: *A conceptual communication graph illustrating the modular, topic-based data flow between nodes in a ROS 2 system.*

1.6.3 Why Use ROS for Our System?

The selection of ROS as the foundational framework for this project is a strategic decision rooted in its extensive ecosystem and proven capabilities. ROS offers a vast collection of open-source packages that support a wide range of robotic functionalities, including navigation, perception, manipulation, and artificial intelligence. These packages significantly accelerate development and minimize redundancy. Furthermore, ROS integrates powerful tools like RViz for visualization and Gazebo for simulation, which allow for efficient testing and prototyping before real-world deployment [3].

Given that our system is designed for an agricultural setting—where automation offers substantial savings in time, energy, and resources—ROS is the ideal foundation for

future integration with a physical robotic platform. The agricultural robotics industry is increasingly adopting ROS due to its adaptability and robust feature set. The download of over 550 million ROS packages in 2023 alone demonstrates its widespread adoption and active community support [4]. By building our system with ROS, we ensure that it is ready for seamless integration with robotic hardware, making the transition from a conceptual model to practical field deployment both smoother and more efficient.

1.7 System-Level Integration: From Environmental Prediction to Robotic Action

1.7.1 Weather Forecasting for Early Disease Prediction

Environmental conditions—notably humidity, temperature, rainfall, and leaf wetness duration—are critical determinants in the development and spread of plant diseases, particularly fungal infections such as late blight in potato crops. Research has established that Phytophthora infestans, the oomycete pathogen responsible for late blight, thrives in moist and cool environments. Optimal conditions for its proliferation are generally around 90% relative humidity and temperatures ranging between 15°C and 25°C. Consequently, weather forecasting serves as an invaluable tool for the early prediction of disease outbreaks.

To this end, several models and decision support systems (DSS) have been developed to forecast disease risk based on meteorological data. A prominent example is BlightCAST [61, 62], a system specifically designed to predict late blight risk by analyzing weather patterns and subsequently issuing regional alerts to farmers. Other systems leverage data from national meteorological agencies or utilize Application Programming Interfaces (APIs) such as OpenWeatherMap and Open-Meteo to assess atmospheric conditions conducive to disease development.

In recent years, machine learning and statistical methodologies have also been increasingly applied to predict disease risk using weather data. These models typically analyze historical weather patterns and corresponding disease occurrences to identify conditions likely to precede an outbreak. However, a common limitation of many such systems is that they are either not crop-specific or lack effective real-time field integration, which can diminish their accuracy and practical utility in dynamic agricultural environments.

Despite their potential, a significant drawback of most existing weather-based forecasting systems is their predominant focus on large-scale regional forecasts. This often results in a failure to provide localized, real-time alerts tailored to the specific microclimatic conditions of individual smallholder farms. Therefore, integrating localized weather forecasting with real-time field data and AI-based disease detection capabilities offers a more robust, precise, and actionable approach to proactive disease management [63].

1.7.2 Automated Spraying, Mobile Robotics, and IoT Interfaces in Smart Farming

The growing demand for sustainable and efficient agricultural practices has spurred the development of automated spraying systems and mobile robotic platforms. These technologies aim to significantly reduce the overuse of chemical treatments while improving the precision and timeliness of interventions. In contrast, traditional farming methods often involve manual pesticide application or uniform spraying across entire fields. Such approaches not only waste valuable resources but also unnecessarily expose agricultural workers to harmful chemicals and increase the overall environmental impact.

Complementing these robotic systems, IoT-based solutions and web interfaces play a pivotal role in enabling effective remote monitoring and control. Through user-friendly dashboards and web applications, farmers can gain real-time access to critical data, including crop health status, prevailing environmental conditions, and the operational parameters of robotic units. Furthermore, these user interfaces can incorporate

advanced features such as alert systems for immediate notifications, historical data logs for trend analysis, and control functionalities for remotely activating or adjusting robotic behavior [64, 65].

While several commercial systems currently offer semi-automated spraying and remote monitoring capabilities, they frequently present limitations. These systems are often expensive, may not be crop-specific, or are generally not designed for efficient small-scale or localized deployment. Therefore, the integration of AI-based disease detection, weather-based forecasting, and precision spraying within a single, cohesive robotic system—all managed through an accessible web interface—presents a more comprehensive and scalable solution for modern agriculture. This is particularly relevant for regions such as Algeria, where the adoption of such advanced agricultural technology is still nascent and its potential largely untapped. Introducing such systems can directly address pressing local challenges by enhancing crop yields to improve food security and by automating manual tasks to alleviate labor shortages.

1.8 Conclusion

This review of the state-of-the-art has revealed that despite significant progress in agricultural robotics, disease detection methodologies, and precision spraying technologies, current solutions often exhibit shortcomings in several key areas. Many existing systems are not specifically designed to address particular crop diseases, such as late blight in potatoes, thereby compromising their accuracy and effectiveness in real-world applications. Furthermore, comprehensive solutions that integrate real-time disease monitoring with weather forecasting and accessible web-based decision-making tools remain largely undeveloped or inaccessible, particularly in regions like Algeria. These limitations underscore a clear and pressing need for an integrated system. By leveraging a state-of-the-art object detection model like YOLOv8, trained on a hybrid, field-relevant dataset, and integrated with localized weather forecasting, it becomes possible to develop a more affordable and farmer-accessible system capable of delivering targeted and timely interventions.

Chapter 2 Proposed Methodology – AI-based Detection of Potato Late Blight

2.1 Domain of Application: Study of Potato Late Blight

Potato (Solanum tuberosum L.) ranks among the five most vital staple food crops globally and is cultivated across a diverse spectrum of climatic conditions. It is characterized by a relatively low water footprint and high nutritional value. These attributes render it a significant crop for Algeria, where both potato production and consumption have expanded considerably over the last three decades. Currently, extensive potato production areas, totaling approximately 90,000 hectares, are established in several regions of this North African nation, and the sector continues to exhibit rapid growth. The majority of potatoes produced in Algeria are consumed domestically, enabling the country to achieve self-sufficiency in potato consumption.

In Algeria, potato cultivation represents the leading vegetable crop in terms of both area and production, with an allocated area of 156,176 hectares yielding approximately 4.6735 million tons, corresponding to an average yield of 29.9 tons/ha as of 2017 [66]. Common cultivation practices include bisecting oversized seed potatoes to reduce costs; these are subsequently planted at a depth of 15 cm, with intra-row spacing of 25-30 cm and inter-row spacing of 65-75 cm [66]. The optimal temperature range for potato tuber formation is 10°C to 16°C (50°F to 60°F) [67]. Potatoes should be planted when soil temperatures range between 7°C (45°F) and 27°C (80°F). Planting in overly moist soil heightens the risk of seed piece decay, whereas excessively cool and dry soil conditions can impede sprouting and emergence. Potato cultivation is compromised when

temperatures deviate beyond these thresholds. Ideal ambient humidity levels for potato plants are reported to be between 60-70% [68, 67].

2.1.1 Challenges in Algerian Potato Production

Potato cultivation in Algeria faces several critical challenges that can be categorized as environmental, technical, and biological (Figure 2.1).

a. *Water Scarcity:*

Algeria faces significant water scarcity due to its limited renewable water resources, with the agricultural sector consuming the majority of available water—over 59% of freshwater withdrawals [69]. Inefficient water management and application practices, especially in agriculture, contribute to substantial water losses and exacerbate the scarcity problem [69]. Inefficient water application practices by farmers frequently result in substantial water losses, particularly in potato production, which is heavily reliant on irrigation.

b. *Mechanization and Labor:*

A low degree of mechanization in potato farming necessitates a significant reliance on manual labor. This dependency poses challenges due to the perceived low status of agricultural work and consequent difficulties in securing labor for critical operations such as planting, harvesting, and irrigation [66].

c. *Plant Diseases:*

Furthermore, plant diseases represent a significant threat to potato crops, capable of causing substantial yield reductions.

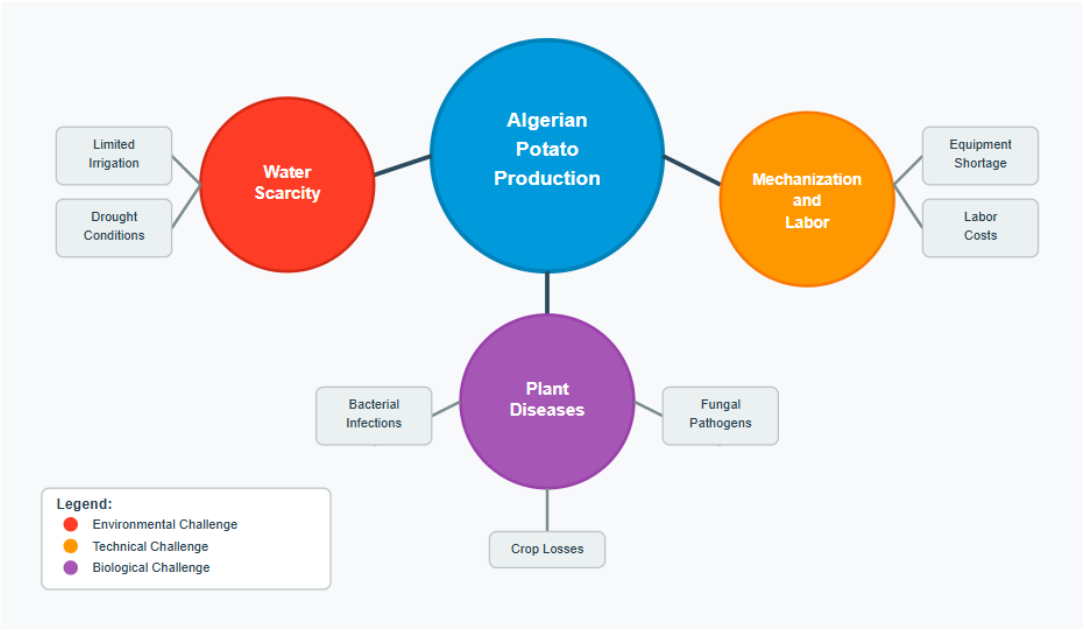


Figure 2.1*: Challenges in Algerian Potato Production*

2.1.2 Overview of Common Potato Diseases

Several diseases commonly affect potato crops, as summarized in (Table 2.1). These include:

- **Brown Rot:** Brown rot (Ralstonia solanacearum), also referred to as bacterial wilt or southern bacterial wilt, affects potato crops across a wide range of warm-temperate, semi-tropical, and tropical zones globally, with occurrences also reported in cooler climates [70].
- **Powdery Mildew:** Powdery mildew (typically caused by Erysiphe cichoracearum or related fungal species) can be a significant foliar disease, particularly prevalent in arid or semi-arid climates [70].
- **Common Scab:** Common Scab (Streptomyces scabies) is present to some degree in most potato-growing regions. It is a major production concern primarily affecting tuber grade quality, with generally minor effects on total yield or storability [70].

- **Potato Leafroll Virus (PLRV):** Potato leafroll, an aphid-transmitted viral disease, is among the most serious diseases affecting potatoes and is responsible for considerable yield losses worldwide wherever potatoes are cultivated [70].

Table 2.1: *Common Potato Diseases*

Disease Name	Causal Agent	Brief	Description Impact
Brown Rot	*Ralstonia solanacearum* (bacterium)	Also known as bacterial wilt; affects potatoes in warm-temperate, semi-tropical, and tropical zones globally, and some cooler climates.	Affects potato crops across a wide range of environments.
Powdery Mildew	*Erysiphe cichoracearum* or related fungal species	A foliar disease that is particularly prevalent in arid or semi-arid climates.	Can be a significant foliar disease.
Common Scab	*Streptomyces scabies* (bacterium)	Present in most potato-growing regions; primarily affects tuber quality.	Major production concern affecting tuber grade quality; generally minor effects on total yield or storability.
Potato Leafroll Virus (PLRV)	Aphid-transmitted virus	An aphid-transmitted viral disease affecting potatoes worldwide.	Among the most serious diseases affecting potatoes; responsible for considerable yield losses worldwide

This research narrows its focus to two particularly impactful foliar diseases: early blight and late blight, with a primary emphasis on late blight due to its devastating potential. A comparative summary of their key features is presented in (Table 2.2).

2.1.3 Early Blight (Alternaria solani)

Early blight, a significant foliar disease of potato (*Solanum tuberosum* L.), is caused by the fungal pathogen *Alternaria solani* Sorauer. It is recognized as one of the most prevalent diseases affecting potatoes and tomatoes. The disease can precipitate major yield losses in most potato-growing regions globally [70, 71].

a. *Historical and Economic Impact:*

Historically, early blight was often regarded as a secondary disease in potatoes, typically causing moderate yield losses under standard growing conditions. However, under conditions conducive to its development (e.g., high humidity, moderate temperatures, and plant stress), early blight can lead to substantial economic losses, especially if significant defoliation occurs before tuber bulking is complete [71]. The primary damage mechanism is premature defoliation, which diminishes photosynthetic capacity and can increase respiration rates in apparently healthy plant tissues. This can result in yield losses ranging from 5% to 50%, with severe infections potentially causing losses between 20% and 50% [72].

b. *Plant and Crop Damage:*

Primary damage manifests as characteristic dark brown to black lesions, often displaying concentric rings (a "target spot" appearance), on the leaves. These lesions can enlarge and coalesce, ultimately leading to leaf necrosis and death. Premature defoliation curtails photosynthesis, resulting in diminished tuber size and overall yield. Disease severity is typically greater on senescent, stressed, malnourished, or physically damaged plants. Environmental factors, including temperature, moisture availability, and leaf wetness duration, significantly influence the severity of early blight [71].

c. ***Symptoms:***

- **Leaves:** Characteristic foliar lesions are dark brown to black, featuring concentric rings that create a "target spot" effect (see Figure 2.2). While typically oval, lesions may remain small and angular under conditions unfavorable for disease progression. These lesions enlarge and coalesce, leading to leaf necrosis [71].
- **Stems:** Lesions can also manifest on stems and petioles. Stem lesions may facilitate the spread of the pathogen to other plant parts [71].
- **Tubers:** Infected tubers exhibit a dry rot characterized by isolated, dark, irregular, and sunken lesions on the surface. Tuber infection usually occurs via wounds, as *Alternaria solani* conidia generally cannot penetrate intact periderm [71].
- **Overall Crop Impact:** Affected plants may display signs of premature senescence, reduced vigor, and, in severe instances, complete vine death [71].

Figure 2.2*: Visual Symptoms of Early Blight on Potato Plants leaf - from plant-doc dataset [53]*

d. *Disease Cause and Spread:*

The presence of free water on leaf surfaces or relative humidity approaching saturation is conducive to spore germination. Germination of *A. solani* spores necessitates a minimum leaf wetness period; studies indicate that spores can germinate at 20°C following a wetting period as short as two hours. Sporulation is often triggered by alternating wet and dry conditions. Leaf wetness duration has been shown to account for up to 90% of the variability in early blight development and severity. Temperature is another critical factor influencing infection, with the optimal range for *A. solani* infection being 20°C to 30°C [71].

e. *Disease Management:*

- **Preventive Measures:** Preventive measures for early blight encompass several cultural and chemical strategies [71]. Crop rotation with non-host crops (e.g., cereals, forage crops) for a period of 3–5 years is recommended to reduce soil-borne inoculum. Careful site selection, ensuring well-drained fields, and thorough sanitation practices, such as the removal of plant debris, help eliminate overwintering pathogen structures. Irrigation scheduling, preferably in the morning, can minimize nocturnal leaf wetness duration. The cultivation of potato cultivars with higher levels of resistance to early blight is an important component of an integrated management strategy. Avoiding plant overcrowding promotes better air circulation, thereby reducing humidity and leaf wetness. Prophylactic application of contact fungicides (e.g., chlorothalonil, mancozeb, copper-based compounds) early in the growing season can be effective. Furthermore, the integration of predictive models for disease forecasting can aid in optimizing fungicide application timing and minimizing their use.
- **Curative and Control Measures:**
  - **Pruning and Removal of Infected Tissues:** The pruning and proper disposal of infected leaves, stems, and plant debris can reduce the pathogen's spore load within the field [71].

- **Environmental Modification:** Where feasible, reducing field moisture levels by improving soil drainage or employing drip irrigation instead of overhead sprinkler systems can mitigate disease spread [71].

- **Biological Control:** Biological control agents, such as *Trichoderma viride* and extracts from *Clerodendrum* spp. leaves, have demonstrated efficacy in reducing early blight severity, especially when integrated with chemical control methods. This approach presents an environmentally sustainable alternative that can help lessen dependence on synthetic fungicides [73].

- **Chemical Control:** Protectant fungicides, such as chlorothalonil, mancozeb, and copper-based formulations, are applied early to prevent initial infection [71, 73]. Following the onset of symptoms, systemic fungicides like difenoconazole, tebuconazole, and flusilazole can be applied to arrest disease progression within the plant [71, 73].

- **Integrated Management:** Integrated management strategies combining biological agents with chemical treatments have shown promising results. For instance, the application of *Clerodendrum* leaf extract or *T. viride* in conjunction with mancozeb demonstrated significant disease control, with reported Percent Disease Index (PDI) values of 29.43% (*Clerodendrum* + 2 mancozeb sprays) and 34.66% (*T. viride* + 2 mancozeb sprays). Such integrated approaches are considered by some researchers to be among the safer options for disease management [73].

2.1.4 Late Blight (Phytophthora infestans)

Late blight, caused by the oomycete *Phytophthora infestans* (Mont.) de Bary, is regarded as one of the most devastating diseases of potato globally, responsible for substantial production losses. The pathogen exhibits high genetic variability and a notable capacity for rapid adaptation to new potato cultivars and fungicide treatments [70, 74]. The damage inflicted by *P. infestans* can be both severe and widespread.

a. *Historical and Economic Impact:*

Late blight was the primary causal agent of the European potato failure in the 1840s, famously leading to the Irish Potato Famine (1845–1852) and the Highland Potato Famine (1846) [74, 75]. Current annual global economic losses attributed to *P. infestans* are estimated at €12 billion, with approximately €10 billion of this impact occurring in developing nations [74, 76]. In the United States, the annual expenditure on fungicides for late blight control alone is approximately $77.1 million, a figure that does not include costs associated with non-fungicidal control measures [74, 76].

b. *Plant and Crop Damage:*

Infected plants display blackened foliage and weakened stems, which can culminate in crop collapse. Under conditions conducive to the disease, an entire potato field can be decimated within a week [74]. Infected tubers develop characteristic reddish-brown to purplish internal lesions. Although initially firm and dry, these infected tubers are highly susceptible to secondary soft rot bacteria, leading to significant losses both in the field and during storage [74]. The rapid and aggressive nature of late blight, particularly under cool, moist conditions, can result in complete crop failure if not managed effectively and promptly [74].

c. *Symptoms:*

Late blight manifests through several distinct symptoms, visually apparent on the leaves, stems, and tubers. These symptoms can develop rapidly, often within 2-3 days post-infection under optimal conditions [74].

- **Leaves:** Initial symptoms on leaves typically appear as water-soaked, irregular, pale green lesions, often near the tips or margins. These lesions rapidly expand into large, brown to purplish-black necrotic areas (see Figure 2.3) [74, 77, 78].
- **White Sporulation:** Under conditions of high humidity, a characteristic white, downy growth, consisting of sporangia (spore-bearing structures) of the pathogen, may be observed on the abaxial (lower) surface of infected leaves, particularly at the lesion margins [74, 77].
- **Stems:** Light to dark brown lesions can develop on stems and petioles, potentially girdling them. Affected stems weaken at these points and may collapse, contributing to the overall blighted appearance of the crop.
- **Overall Crop Impact:** Under conditions favorable for the pathogen, the entire crop can be destroyed within a short period, sometimes as quickly as one week [74, 77].
- **Tubers:** Tubers become infected when sporangia, washed from diseased foliage by rain or irrigation water, infiltrate the soil. Infected tubers exhibit irregular, reddish-brown to purplish discolored areas that extend into the flesh. Initially, these affected tissues are firm and dry, but they are highly susceptible to secondary invasion by soft rot bacteria, leading to tuber decay in the field or during storage [74, 77].

Figure 2.3: *Visual Symptoms of Late Blight on Potato Plants leaf - from plant-doc dataset [53]*

d. *Disease Cause and Spread:*

Phytophthora infestans is an oomycete, colloquially known as a water mold, and is the causal agent of potato late blight. Oomycetes are characterized by different spore types, each fulfilling a specific role in the disease cycle. The pathogen primarily disseminates via sporangia, which are adapted for aerial dispersal over longer distances or can be spread by water splash within a field [74, 78]. Under conditions of high moisture, such as pooled water on soil surfaces or persistent leaf wetness, sporangia can germinate indirectly by releasing motile zoospores. These zoospores, capable of swimming for approximately 2-10 hours, facilitate short-distance dispersal and initiate new infections within the crop canopy [78]. High humidity levels (90-100%) coupled with moderate temperatures (12-23°C) are optimal for sporangia production. Direct germination of sporangia and subsequent infection typically occur at temperatures between 17-23°C, while zoospore production and infection are favored by cooler temperatures, ranging from 6-17°C [78].

e. *Disease Management:*

- **Preventive Measures:** Effective late blight management hinges on strategies designed to minimize initial inoculum sources and prevent the subsequent development of secondary inoculum on host plants. Several practices contribute to late blight control [78]. These include planting only certified, disease-free seed tubers; destroying cull piles and waste potato tubers, which can harbor the pathogen; eliminating volunteer potato and tomato plants that may arise from previous plantings; eradicating unmanaged or abandoned infected host plants in the vicinity; optimizing row spacing to enhance airflow and reduce canopy moisture; cultivating resistant potato cultivars; and applying fungicides prophylactically to prevent infection establishment.

- **Curative and Control Measures:**

  - **Biological Control:** Certain natural antagonists, including *Trichoderma viride*, *Bacillus subtilis*, and *Pseudomonas fluorescens*, have shown potential in suppressing the growth of *P. infestans*. Their mechanisms of action include competition for nutrients, production of inhibitory compounds (antibiosis), or induction of host plant resistance [79, 80].

  - **Sanitation and Removal of Infected Material:** Prompt removal and destruction of infected leaves, stems, and tubers are crucial to reduce the inoculum source within the field. Composting of infected plant material should be avoided as it may not effectively eliminate the pathogen and could contribute to its spread [80, 81].

  - **Environmental Modification:** Cultural practices aimed at reducing humidity and leaf wetness duration can impede disease development. These include ensuring adequate plant spacing and selective pruning to improve airflow within the canopy. Avoiding overhead irrigation in favor of methods that minimize leaf wetness (e.g., drip irrigation) is also beneficial.

- **Chemical Control:** Systemic fungicides, including active ingredients such as metalaxyl, cymoxanil, mandipropamid, and propamocarb, are effective in arresting the progression of late blight. These fungicides can penetrate plant tissues by offering a degree of curative action if applied shortly after infection has occurred. The efficacy of curative sprays is highest when applied at the very first indication of symptoms. Repeated applications are often necessary, with frequency dictated by disease severity and prevailing environmental conditions [80].

- **Efficacious Chemical Control Strategies:** Fungicides containing cyazofamid and mandipropamid have demonstrated high efficacy, particularly when applied preventively before anticipated high-risk infection periods. Research conducted in Denmark indicated that the application rates of these fungicides could be reduced by up to 30% by tailoring dosages based on the host cultivar's resistance level and prevailing disease pressure [74, 80]. The combination of plant activators, such as β-aminobutyric acid (BABA), with protectant fungicides like mancozeb has been reported to be more effective than either product used alone. For instance, a specific ratio of 5 parts BABA to 1 part mancozeb exhibited synergistic effects in controlling late blight [80].

Table 2.2: *Comparative Summary of Early Blight vs. Late Blight.*

Feature	Early Blight (Alternaria solani)	Late Blight (Phytophthora infestans)
Causal Organism	Fungal pathogen (Alternaria solani)	Oomycete (water mold) (Phytophthora infestans)
Typical Leaf Symptom	Dark brown to black lesions, often with concentric rings ("target spot" appearance); lesions enlarge and coalesce, leading to necrosis.	Initially water-soaked, pale green, irregular lesions (often near tips/margins), rapidly becoming large, brown to purplish-black necrotic areas. White, downy mildew (sporangia) on lower leaf surfaces in high humidity.
Tuber Symptom	Dry rot; isolated, dark, irregular, sunken lesions on the surface. Infection typically occurs via wounds.	Reddish-brown to purplish discolored areas extending into the flesh; initially firm and dry, but very susceptible to secondary soft rot bacteria, leading to decay.
Conditions Favoring Disease	High humidity (near saturation for spore germination), moderate temperatures (optimal 20°C-30°C for infection), leaf wetness (min. 2 hours at 20°C for spore germination); alternating wet/dry conditions for sporulation. Stressed or senescent plants are more susceptible.	High humidity (90-100%), moderate temperatures (12-23°C for sporangia production; 17-23°C for direct germination; cooler 6-17°C for zoospore production/infection), persistent leaf wetness or pooled water for zoospore release.
Relative Economic Impact	Can cause significant yield losses (5-50%, potentially 20-50% in severe cases if defoliation occurs before tuber bulking). Historically often considered secondary but can cause substantial economic losses.	Considered one of the most devastating potato diseases globally. Historically caused the Irish Potato Famine. Current global annual losses estimated at €12 billion. Can lead to complete crop failure rapidly.

2.2 Proposed Solution Architecture

2.2.1 General System Architecture

The proposed system architecture is designed around a central decision-making module that serves as the core intelligence of the robotic platform. This module is responsible for managing and processing all incoming data to determine if a threat to the potato crop exists. As illustrated in (Figure 2.4), the decision-making module receives two primary inputs. The first is from the AI-based visual detection model, specifically the YOLOv8 model, whose selection was justified in (Section 1.4.3), which provides real-time identification of disease symptoms. The second input is from the Weather-Based Prediction System, which assesses environmental conditions to forecast disease risk. Based on the fused information from these two streams, the decision-making module generates a single, precise output: a command to activate the precision spraying system.

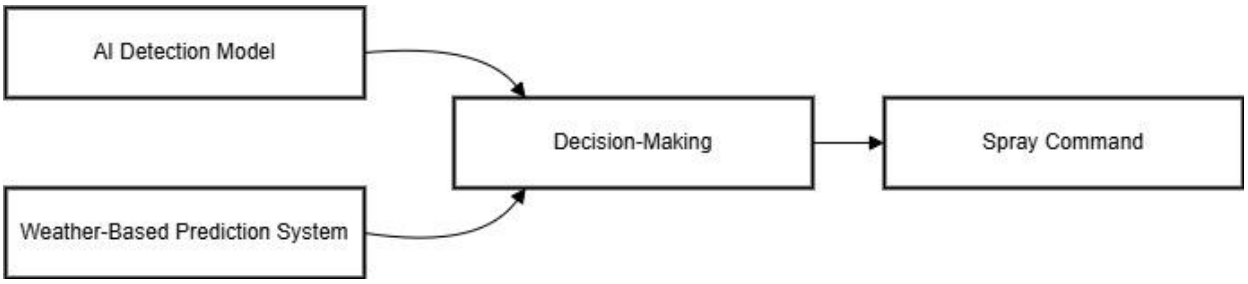


Figure 2.4: *Block diagram illustrating the core system logic*

2.2.2 Overview of Subsystems:

a. *AI Detection Module (YOLO-based)*

The **AI Detection Module** is a critical component that significantly enhances the system's power and efficacy. Its primary function is the real-time detection of late blight infections from visual symptoms present on the plant foliage.

This automated approach offers key advantages over manual human inspection by identifying subtle signs of disease at very early stages, which are often missed by the

human eye, and by accurately distinguishing late blight from other visually similar diseases, thereby reducing the risk of misdiagnosis.

The module operates by processing a real-time video stream from the robot's camera, following the algorithmic process outlined in (Figure 2.5). Each frame is analyzed to generate detection data. Upon positively identifying a disease infection, the module sends both a confirmation command and the corresponding image with detection data to the decision-making logic for further processing.

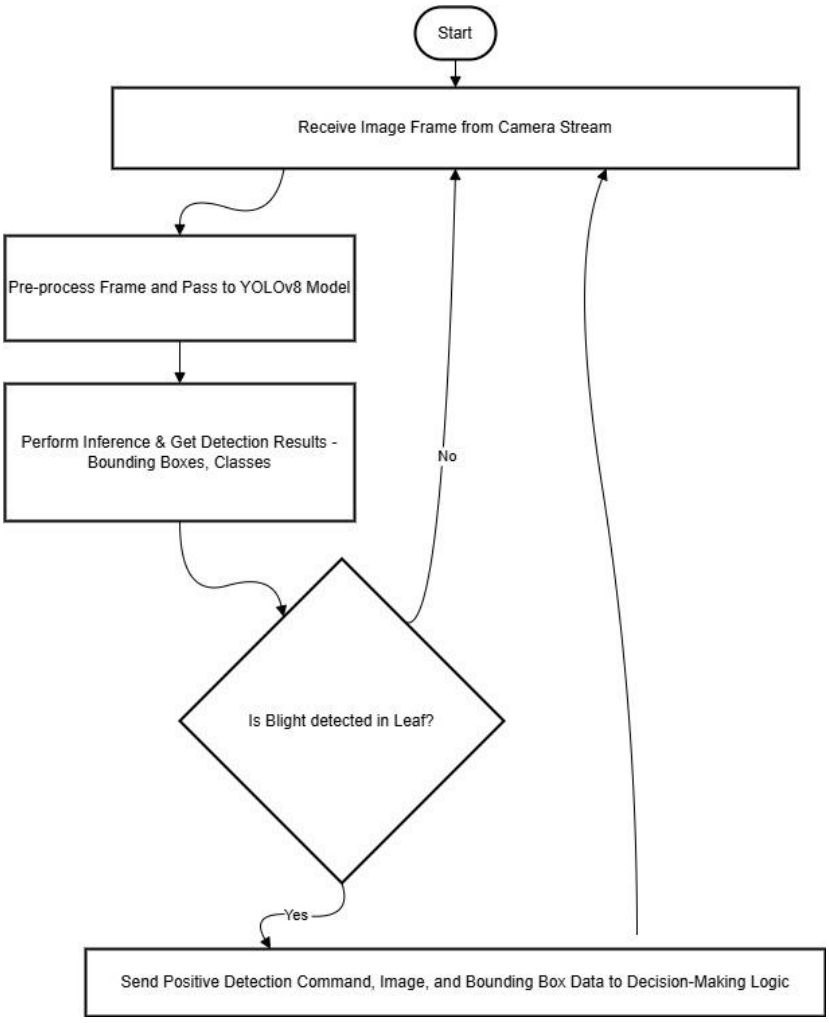


Figure 2.5: *Algorithmic flowchart illustrating the operational steps of the AI Detection Module.*

b. *Weather-Based Prediction System*

To minimize crop losses, the system's architecture prioritizes proactive disease management over reactive treatment. Central to this strategy is the integration of a weather-based prediction system, designed to forecast and mitigate threats before they emerge. The system operates on a continuous 10-minute cycle, as illustrated in (Figure 2.6). In each cycle, it fetches both the current weather and a 48-hour forecast from meteorological APIs such as OpenWeatherMap [82] and Open-Meteo [83].

The core of the algorithm involves analyzing the upcoming 24-hour forecast to identify two critical windows: periods where conditions are favorable for the emergence of late blight and optimal times for pesticide application. Based on this analysis, the system publishes a comprehensive summary that includes the current weather, the next predicted high-risk period for blight, the next suitable time for spraying, and a general 24-hour forecast. This proactive approach, which combines real-time data with predictive insights, enables timely and precise interventions before a full-scale outbreak can occur.

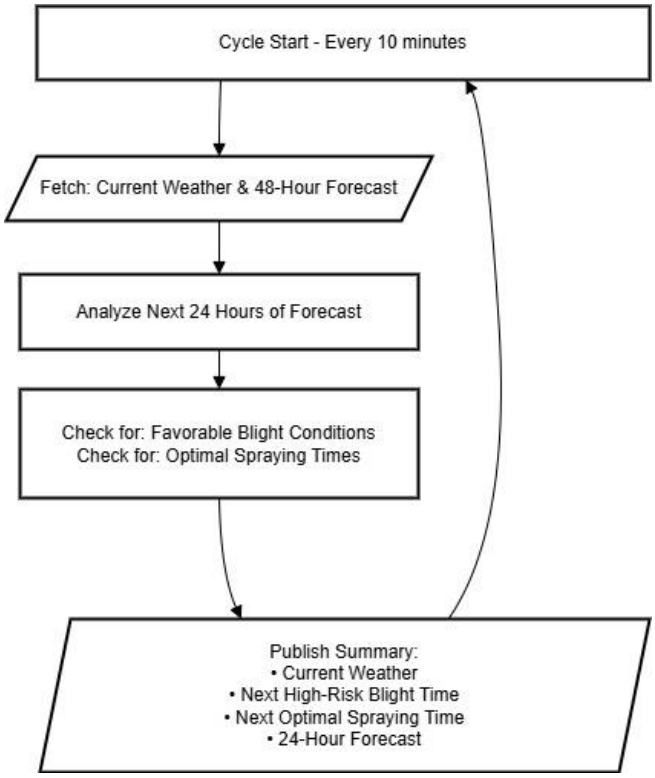


Figure 2.6: *Flowchart of the Weather-Based Prediction System's Operational Cycle.*

c. *Decision-making Strategy*

The system's decision-making logic, illustrated in (Figure 2.7), manages and automates pesticide spraying based on a combination of weather conditions and real-time disease detection. This logic is composed of two main components: a "Spray Scheduler" and a "Spray Decider," whose core algorithms are detailed in the flowcharts in (Figure 2.8).

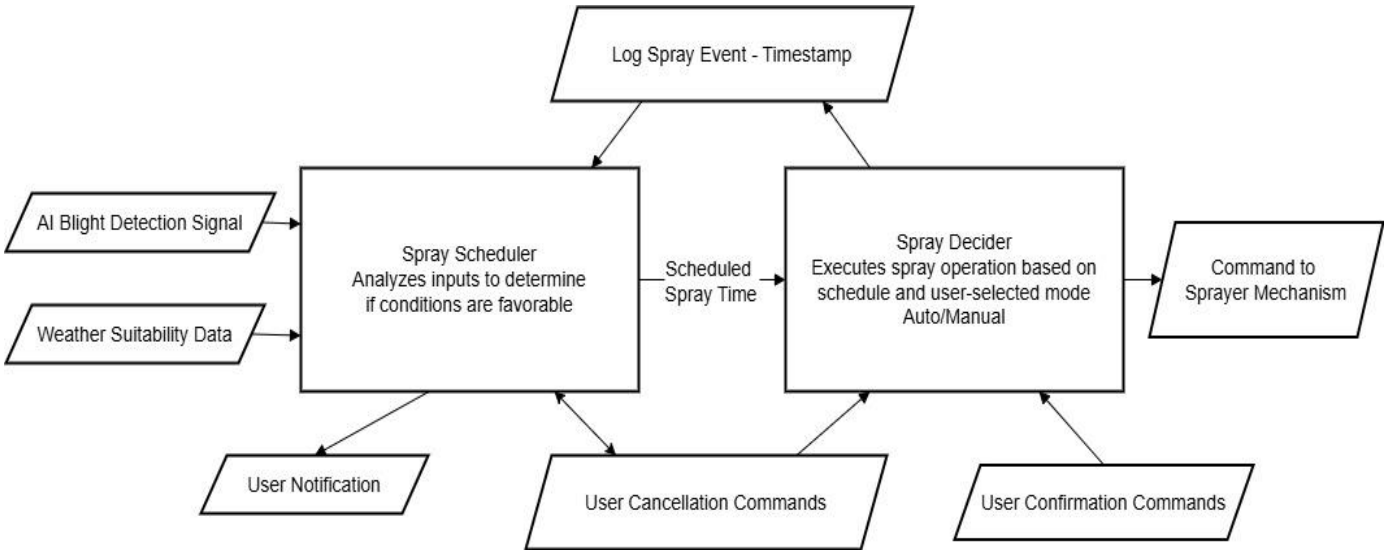
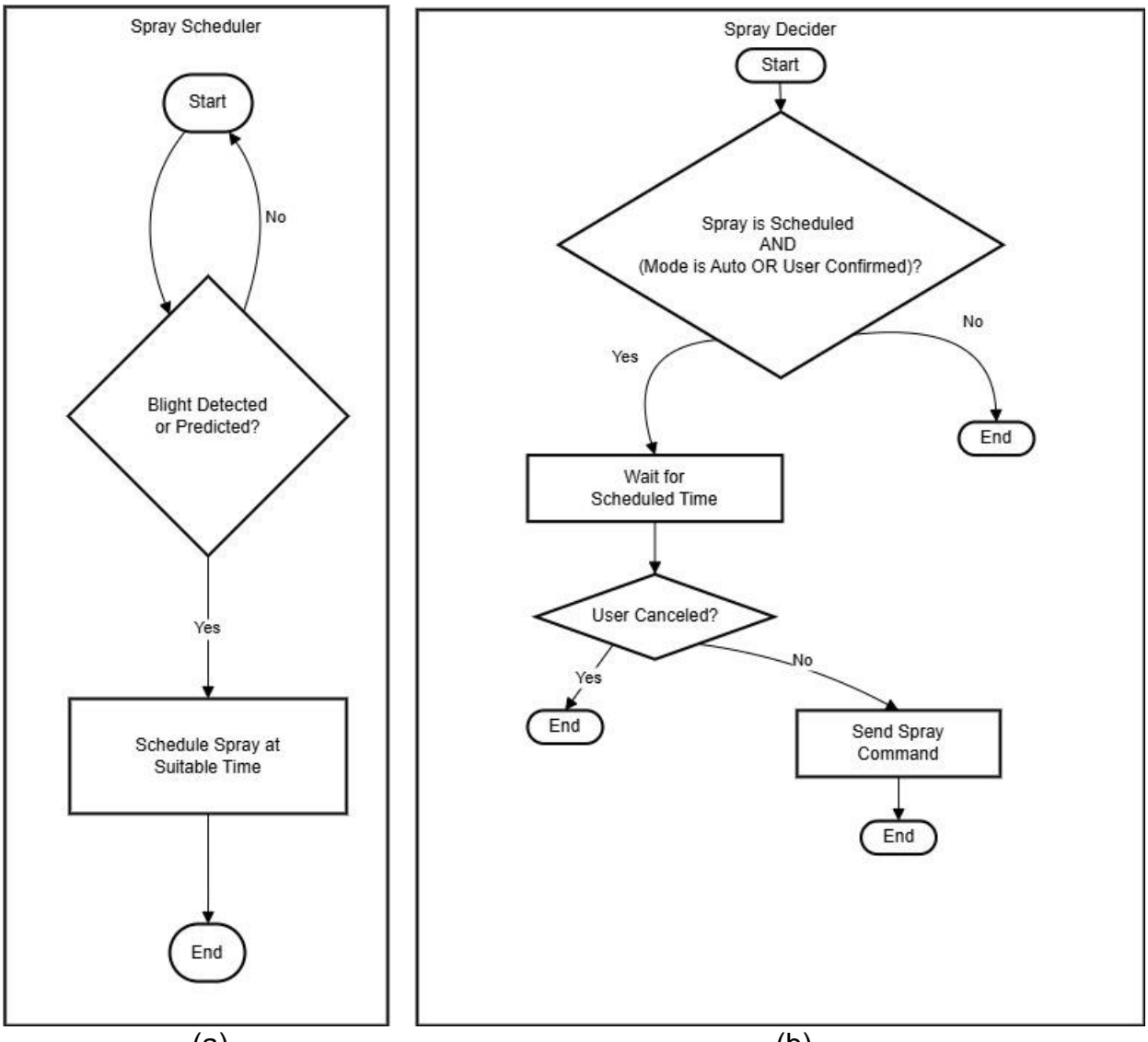


Figure 2.7: *Architecture of the Decision-Making Strategy, detailing the data flow between the Spray Scheduler and Spray Decider components.*

The **Spray Scheduler** receives weather suitability data from the weather-based prediction system, blight detection signals from the AI detection module, logs of previous spray events, and any cancellation signals from the user. It processes this information to determine if conditions are favorable for spraying. If they are, it schedules a spray for a specific time and sends this schedule to the Spray Decider, while also sending a notification to the user.

The **Spray Decider** is the component responsible for executing the spraying operation. It monitors for scheduled spray times, user confirmations, and cancellation commands. When a scheduled time arrives, it will either start spraying automatically or wait for user approval, depending on the selected operational mode. Once spraying begins, it records

the start and completion times of the event. The system supports two modes: an **automatic mode**, where spraying starts at the scheduled time unless canceled, and a **manual mode**, where spraying only proceeds after the user explicitly accepts the proposed schedule. This coordinated design ensures that spraying decisions are efficient, timely, and safe.



(a) (b)

Figure 2.8: *Algorithmic flowcharts for the decision-making logic, showing (a) the Spray Scheduler and (b) the Spray Decider.*

2.3 Vision System

The development of the vision system was an iterative process focused on optimizing both the dataset and model training parameters to achieve robust, real-world performance. This section details the experimental journey, from initial baseline models to the final, deployed solution.

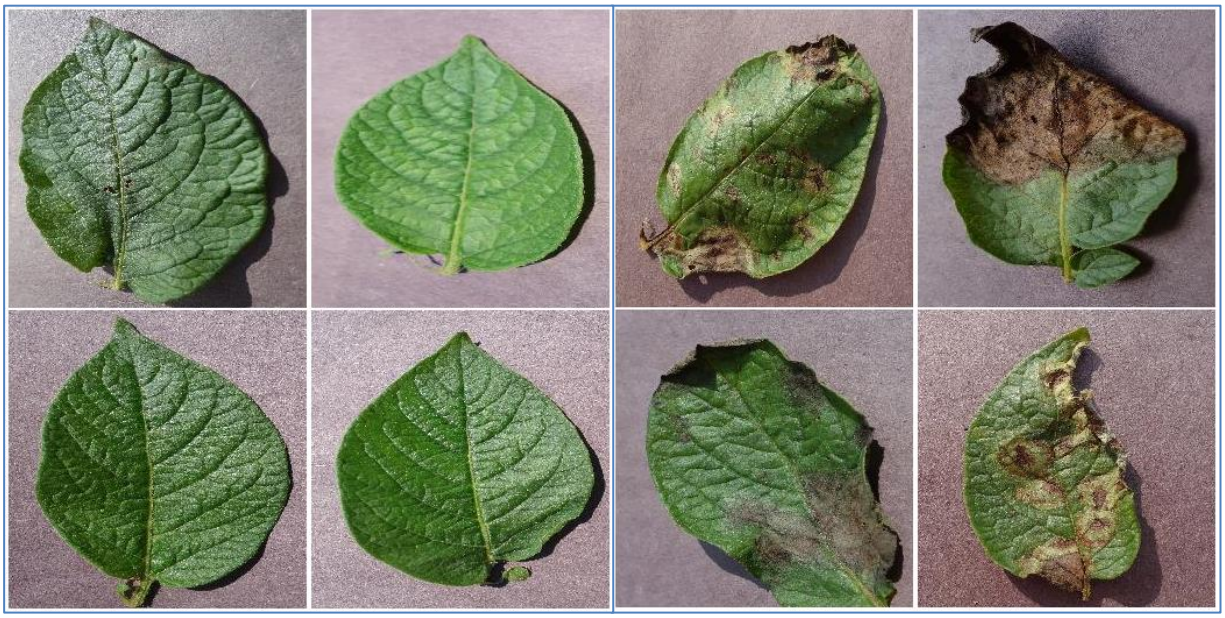
2.3.1 Dataset Collection and Labeling: An Iterative Approach

The creation of a suitable dataset was one of the most challenging aspects of this project. The final dataset was a composite, compiled from various open-source datasets available on the internet, to ensure diversity.

All images were consolidated and managed on the Roboflow platform, which facilitated manual labeling. The process involved defining a set of classes and then drawing bounding boxes around the relevant features in each image. After labeling, a series of data augmentation techniques—including rotations, flips, blurs, and contrast adjustments—were applied to increase the dataset's variability. For each dataset version, a stratified 80:10:10 split was used to create the training, validation, and test sets. The platform allowed for the dataset to be exported directly in the YOLOv8 format, providing a code snippet for easy integration into the training script. The primary experimental variables in this stage were the choice of source images and the definition of the classes used for detection. Several distinct datasets were created and tested, with the results detailed in **Table 2.3**.

- **First Dataset Version (Dataset-V1):** The initial experiment aimed to classify leaves as either infected or healthy. For this, two classes were created: "blight_leaf" and "h_leaf". The dataset was sourced exclusively from the PlantVillage collection (see Figure 2.9), consisting of 1000 images of infected leaves and all 152 available images of healthy leaves. After training, the results were very poor due to catastrophic overfitting. This failure was primarily caused by the severe class imbalance in the

dataset; with significantly more infected than healthy samples, the model learned to default to detecting blight, rendering it incapable of accurate differentiation.



(a) (b)

Figure 2.9: *Example images from the PlantVillage dataset showing (a) healthy potato leaves and (b) leaves infected with late blight, both on uniform backgrounds.*

- **Second Dataset Version (Dataset-V2):** To address the issues from the first version, the approach was fundamentally changed. Instead of classifying healthy versus infected leaves, this version focused purely on detection. A new single-class dataset was created using 1000 images of blight-infected leaves from PlantVillage, with the class labeled "blight-leaf". While this represented a step forward, as the model could now identify infections, it introduced a new, significant problem: the model began to misclassify soil as blight. This was because the PlantVillage dataset consists of images with clean, uniform backgrounds, and the model had not learned to distinguish blight from the complex textures found in a real-world soil environment.

- **Third Dataset Version (Dataset-V3):** To solve the soil misclassification issue, a hybrid dataset was created. The number of PlantVillage images was reduced, and more real-world images from the Plant-Doc dataset (see Figure 2.10) were added to expose the model to natural environments. The labeling strategy was also refined to be more granular, using three classes: "blight" for the specific infected area, "leaf" for the entire leaf, and "dead_leaf" to account for the senescent tissue frequently observed during labeling. Although this was a time-consuming manual annotation process, the goal was to teach the model more precise distinctions. However, the training results revealed a critical flaw in this approach: the model learned to misclassify soil as "dead_leaf", while also missing a significant number of true blight infections.

Figure 2.10: *Example images from the Plant-Doc dataset showing late blight symptoms under various real-world field conditions.*

- **Intermediate Two-Class Versions (Dataset-V4, V5, V6):** Based on the previous results, the strategy shifted to a two-class system ("blight", "leaf") to improve precision by first identifying the leaf region and then detecting blight only within that region. Several versions of this approach were tested. Dataset-V4 applied this strategy to the PlantVillage-only image set. Dataset-V5 and its improved version, Dataset-V6 (composed of 91 Plant-Doc images and a balanced mix of 152 healthy and 136 blighted PlantVillage images), used a hybrid approach. However, all these versions exhibited significant flaws: the model tended to incorrectly label the entire image as "leaf", a failure likely caused by overfitting due to the limited dataset size. More

critically, the misclassification of soil as "blight" persisted, indicating the model still struggled with domain shift and could not generalize to real-world backgrounds.

- **Seventh Dataset Version (Dataset-V7):** In parallel, further experiments were conducted to refine the single-class approach. Dataset-V7 was a single-class ("blight-leaf") hybrid dataset, combining a reduced number of PlantVillage images with real-world Plant-Doc images. While an improvement, it still struggled to detect small, early-stage infections.

The persistence of issues across so many dataset versions suggested that the dataset composition was not the only factor limiting performance. It became clear that to isolate the problem and find an optimal solution, it was also necessary to systematically evaluate the model's training parameters.

2.3.2 Model Training

a. *Experimental Trials and Observations:*

All model training and evaluation experiments were conducted within the Google Colaboratory environment (Colab) to leverage its free temporary access to powerful NVIDIA Tesla T4 GPUs. The training script was built primarily around the **ultralytics** library, which provides pre-trained YOLO models and a streamlined training pipeline, and the **roboflow** library for direct dataset downloading.

The training process was initiated using the **ultralytics** Command Line Interface (CLI), which allows for efficient model training with a single command. The experimental methodology focused on comparing the performance of two lightweight YOLOv8 model sizes, **YOLOv8n** (nano) and **YOLOv8s** (small), as these are better suited for deployment on resource-constrained hardware. The primary hyperparameter adjusted between experiments was the number of training epochs.

- **First Training Attempt:** The initial training run utilized the **YOLOv8s** model with its default configuration on Dataset-V1. As shown in **Table 2.3** (ID 1), training on this

highly imbalanced dataset resulted in catastrophic overfitting. The model learned to classify nearly every leaf as blight, rendering it completely ineffective for reliable detection and yielding poor results. This underscored the critical need for a more balanced and contextually diverse dataset for subsequent versions.

- **Systematic Parameter Tuning:** After numerous experiments with various datasets using the **YOLOv8s** model and a high epoch count (100) still yielded suboptimal results, as detailed in **Table 2.3**, a new research insight emerged. It was hypothesized that the high number of epochs might be contributing to overfitting, especially for the more focused two-class detection task. Furthermore, the impact of model size had not been systematically evaluated. Therefore, a second phase of more focused experiments was designed to isolate the optimal parameters. For this phase, the epoch count was significantly reduced to 25, and both the **YOLOv8s** and **YOLOv8n** models were trained on the most promising dataset candidates (V2, V3, V6, and V7). The goal of this comparative analysis was to determine the best combination of dataset structure, model size, and training duration for the specific task of real-world blight detection.

- **Analysis of Comparative Training Results (IDs 8-15):** The second phase of experiments, detailed in **Table 2.3** (IDs 8-15), compared the **YOLOv8s** and **YOLOv8n** models across the most promising dataset versions (V2, V3, V6, and V7) with a reduced training duration of 25 epochs. This analysis revealed critical trade-offs between model size, dataset structure, and real-world performance. The results from datasets V2 (IDs 14, 15) and V3 (IDs 12, 13) were consistently poor for both

- model sizes. Despite achieving high mAP scores in some cases (e.g., 0.934 for **YOLOv8n** on V2), the models failed validation on real images, persistently misclassifying soil as blight. This demonstrated that a high metric score is meaningless if the underlying dataset lacks real-world diversity. The single-class hybrid dataset, V7, showed some promise. The **YOLOv8s** model (ID 8) proved capable of detecting small blight infections, but at the cost of missing some instances when soil was present. The **YOLOv8n** version (ID 9) had good general accuracy but

missed a significant amount of blight and still incorrectly detected soil. The most successful outcomes were achieved with the two-class hybrid dataset, V6. The **YOLOv8s** model (ID 10) achieved a high mAP@0.5 of 0.893 but failed to detect smaller blight spots. In direct comparison, the **YOLOv8n** model (ID 11), while having a slightly lower mAP@0.5 of 0.874, provided qualitatively better results. It demonstrated good overall detection and accuracy on real images, successfully distinguishing leaves from the background and detecting a majority of blight infections without the critical flaw of misidentifying soil. While neither model was perfect at detecting the smallest lesions, the superior generalization and reliability of the **YOLOv8n** model in Experiment 11 made it the clear choice for the final deployed system (see Figure 2.11).

b. ***Explanation of Evaluation Metrics:***

To interpret the above results accurately, the following definitions and formulas apply:

mAP@0.5 (mean Average Precision at IoU=0.5) measures the average precision across all classes using a threshold of 0.5 for the Intersection over Union (IoU). A detection is considered correct if the IoU between the predicted and ground-truth bounding boxes is at least 0.5. It is calculated as:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i$$

where N is the number of classes, and APi is the average precision for class i.

Precision reflects the accuracy of positive predictions:

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

where:

- **TP (True Positives)**: Instances correctly identified as positive (e.g., model predicted blight, and it was indeed blight).
- **FP (False Positives)**: Instances incorrectly identified as positive (e.g., model predicted blight, but it was not blight).

Recall measures the ability to find all relevant instances:

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

where:

- **FN (False Negatives)**: Positive instances that the model failed to detect (e.g., there was blight, but the model did not detect it).

The results indicate that the model was highly effective at detecting both blight-infected and healthy leaves. Notably, the blight class achieved a high precision (0.894), indicating very few false detections, while the leaf class showed excellent recall (0.895), meaning most relevant instances were correctly identified.

(a) (b)

Figure 2.11: *Qualitative comparison of detection results. (a) The model from ID 10 fails to detect smaller lesions. (b) The final model from ID 11 shows accurate and reliable detection.*

Table 2.3: *Comprehensive Results of Iterative Experiments Comparing Dataset Configurations, YOLOv8 Model Variants, and Training Parameters*

ID	Dataset version	Dataset Source	Total Image	Class Labels	YOLO Model Type	Epochs	mAP @0.5	mAP@ 0.5:0.95	Precision	Recall	test on real images
1	V1	152 from plantvillage without blight 1000 from plantvillage with blight	1152	2 classes: blight_leaf, h_leaf	yolov8s	100	0.678	0.546	1.000	0.356	it detects everything as blight leaf
2	V2	1000 from plantvillage with blight	1000	1 class: blight-leaf	yolov8s	50	0.684	0.342	0.772	0.582	the detection is bad the accuracy is not bad it misses a lot it detects soil as blight
3	V3	15 from plant-doc 30 from plantvillage with blight	45	3 classes: blighte, dead-leaf, leaf	yolov8s	100	0.881	0.629	0.948	0.816	the detection is bad miss some leaf miss a lot of blight detect soil as blight and dead leaf
4	V4	1000 from plantvillage with blight	1000	2 classes: blight, leaf	yolov8s	100	0.891	0.761	0.836	0.896	it detects the hole image as leaf the accuracy is bad it detects soil as blight
5	V5	225 from plantvillage without blight 100 from plantvillage with blight	325	2 classes: blight, leaf	yolov8s	100	0.875	0.657	0.860	0.871	it detects the hole image as leaf the accuracy is bad it detects soil as blight

6	V6	91 from plant-doc 152 from plantvillage without blight 136 from plantvillage with blight	371	2 classes: blight, leaf	yolov8s	100	N/A	N/A	N/A	N/A	it detects the hole image as leaf the accuracy is bad it detects soil as blight
7	V6	91 from plant-doc 152 from plantvillage without blight 136 from plantvillage with blight	371	2 classes: blight, leaf	yolov8n	100	N/A	N/A	N/A	N/A	the detection is good average accuracy the detection of leaf is pretty good
8	V7	104 from plant-doc 100 from plantvillage with blight	204	1 class: blight-leaf	yolov8s	25	0.691	0.35	0.703	0.658	the detection is pretty good the accuracy is good it's detected small blight infection its miss some blight when soil is present
9	V7	104 from plant-doc 100 from plantvillage with blight	204	1 class: blight-leaf	yolov8n	25	0.677	0.334	0.759	0.587	good detection is good accuracy its miss a lot of blight it detects soil as blight
10	V6	91 from plant-doc 152 from plantvillage without blight 136 from plantvillage with blight	371	2 classes: blight, leaf	yolov8s	25	0.893	0.716	0.846	0.867	Average detection it’s not detected small blight
11	V6	91 from plant-doc 152 from plantvillage without	371	2 classes: blight, leaf	yolov8n	25	0.874	0.704	0.905	0.833	the accuracy and detection are good it detects a lot but not all blight

		blight 136 from plantvillage with blight									it’s not detected small blight
12	V3	15 from plant-doc 30 from plantvillage with blight	45	3 classes: blighte, leaf, dead-leaf	yolov8s	25	0.905	0.542	0.886	0.833	the detection is bad miss a lot of leaf and blight detect soil as blight its didn’t detect dead leaf
13	V3	15 from plant-doc 30 from plantvillage with blight	45	3 classes: blighte, leaf, dead-leaf	yolov8n	25	0.869	0.576	0.837	0.840	the detection is bad the accuracy is bad it detects soil as blight-leaf
14	V2	1000 from plantvillage with blight	1000	1 class: Blight	yolov8s	25	0.928	0.514	0.942	0.756	the detection is bad the accuracy is bad it misses a lot it detects soil as blight
15	V2	1000 from plantvillage with blight	1000	1 class: Blight	yolov8n	25	0.934	0.519	0.914	0.872	the detection is bad the accuracy is bad it detects soil as blight

2.3.3 Conclusions from Experimental Analysis for Final Model Training

The series of experiments yielded several key insights that will directly inform the methodology for training the final vision system model. The analysis concluded that achieving optimal performance requires enhancements in both the dataset and the training parameters.

The primary conclusion regarding the dataset is that a larger and more diverse set of images is necessary. To improve precision and reduce the persistent issue of false positives where the model misidentifies soil, the final training dataset must be enhanced with a significant number of negative samples. This includes images of healthy plants in field conditions, bare soil, and random images of other common field elements to teach the model what to ignore. This will compel the model to learn more discriminative features specific only to late blight.

Regarding the training configuration, the comparative analysis demonstrated that the **YOLOv8n** model is the most suitable choice for this specific task. Its smaller capacity proved to be an advantage, making it less prone to overfitting on a focused, two-class problem compared to the larger **YOLOv8s** model. Finally, the experiments clearly indicated that a lower number of training epochs is more effective; setting the epoch number to 25 provided the best balance between performance and training efficiency, avoiding the diminishing returns of longer training runs. These conclusions form the basis for the final model training protocol.

2.3.4 Final Model Training and Evaluation Results

Based on the conclusions drawn from the iterative experimental phase, a final, optimized dataset was constructed and used to train the selected YOLOv8n model. This section details the configuration of the final dataset, the specific training parameters used, and the resulting performance metrics.

a. *Final Dataset Configuration*

The final dataset was strategically compiled to maximize diversity and real-world applicability, directly addressing the critical issues of domain shift and class imbalance identified in earlier versions. A pivotal decision was to create a balanced combination of image sources. This included exactly 410 real-world images from the Plant-Doc dataset, which were essential for exposing the model to natural environmental variability. Because the Plant-Doc images often contain multiple leaves, and for better balance, this was complemented by a set of 1000 high-quality, realistic images with blight from the PlantVillage collection. To further ensure enough healthy leaf examples for detection, we also added 200 images without infection from PlantVillage, ensuring the model had clear, classifiable examples of both infected and healthy tissue.

To further enhance the dataset's diversity, 80 images from different neighboring areas were provided by INPV (Institut National de la Protection des Végétaux) during a field visit, as exemplified in (Figure 2.12). Additionally, 10 self-captured images of potato plants grown at home were included, such as those shown in (Figure 2.13). Crucially, all these images were initially labeled using the best model from previous tests to identify and correct any labeling issues. We also employed "hard negative mining" by using the model to detect and save images from environments without any leaves or blight. From these, 500 images where the model made a detection but no real object was present were collected and added as negative images to the dataset, as illustrated in (Figure 2.14). This process was a key step in teaching the model what to ignore, thus reducing the false positive detections that plagued earlier experiments. This comprehensive approach resulted in a base dataset of 2200 images across two final classes: "blight" and "leaf". The distribution of these two classes within the dataset was carefully managed to be very close, aiming for balanced representation, as further detailed in (Figure 2.15).

The dataset was manually partitioned, to ensure the best training, into a 96%/3%/1% split, resulting in 2108 images for the training set, 69 for the validation set, and 23 for the test set. To further enhance robustness and simulate real-world conditions, a

comprehensive set of augmentations was applied, effectively tripling the dataset size to 6378 images.

Figure 2.12: *Sample images provided by INPV*

Figure 2.13: *Sample images of potato plants grown at home*

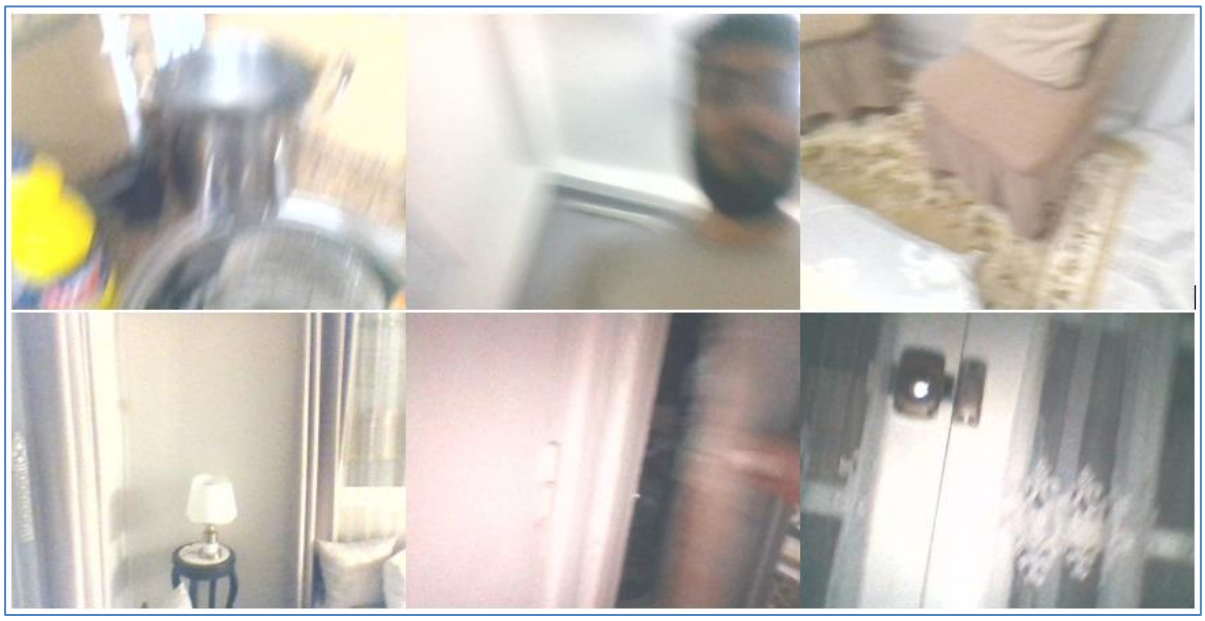


Figure 2.14: *Sample of negative images used for hard negative mining*

COLOR	CLASS NAME	COUNT
	blight	8 307
	leaf	8 064

Figure 2.15: *Class distribution in the dataset*

b. *Training Setup*

The final training configuration was informed by the experimental analysis. We selected the lightweight YOLOv8n model for its superior real-world generalization and lower risk of overfitting. Based on previous results, the training was set for an optimal duration of 25 epochs, and an input image size of 800x800 pixels was used to improve the detection of small, early-stage lesions.

c. *Evaluation Results*

The final trained model demonstrated strong performance across all key metrics during both the training and validation phases. The model architecture consists of 168 layers and approximately 3 million parameters. A detailed breakdown of the model's performance is illustrated by the confusion matrix (Figure 2.16) and a series of performance evaluation curves (Figure 2.17).

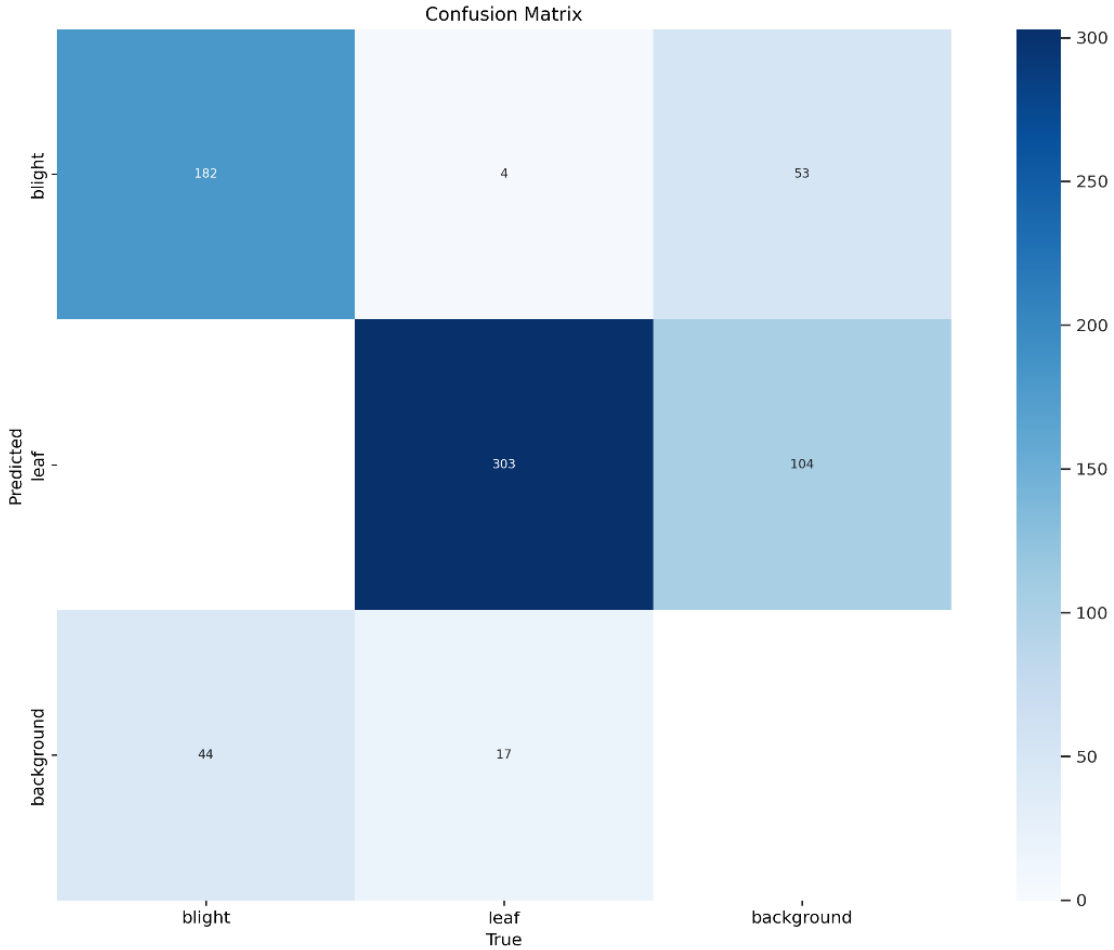


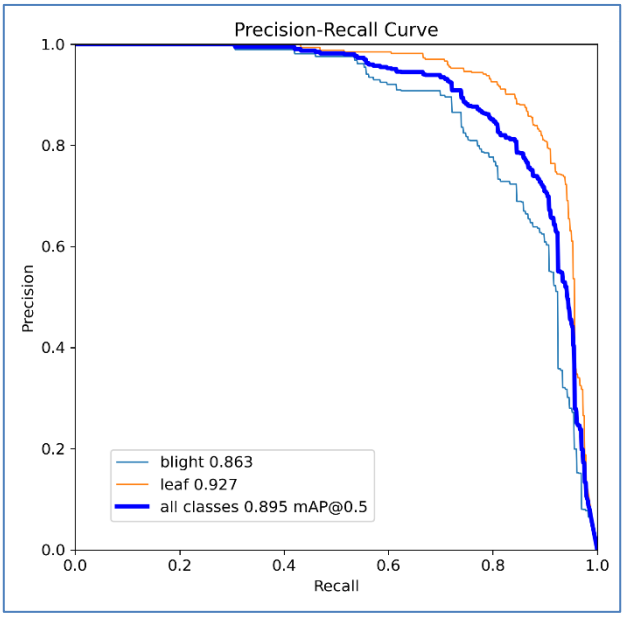
Figure 2.16: *Confusion matrix for the final YOLOv8n model.*

Training Phase Results: The model achieved excellent results on the training set, indicating successful learning.

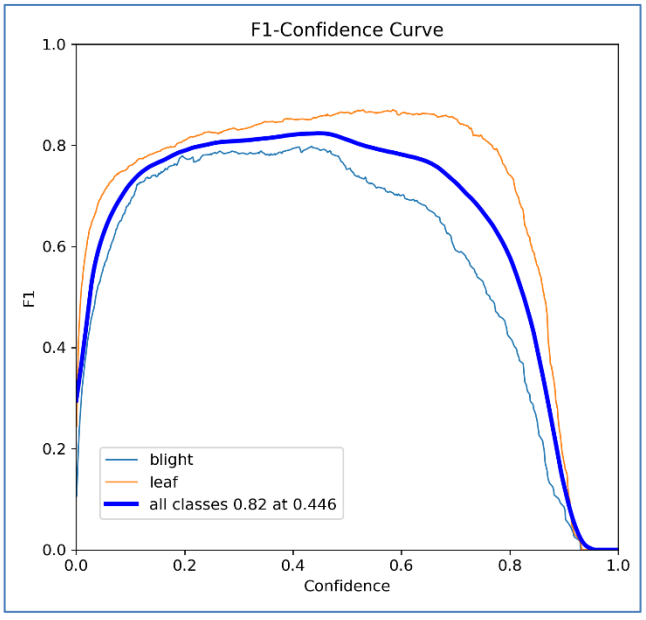
- Overall mAP@0.5: 0.895
- Blight Class mAP@0.5: 0.862 (with a precision of 0.895 and recall of 0.716)
- Leaf Class mAP@0.5: 0.928 (with a precision of 0.820 and recall of 0.895)

Validation Phase Results: Critically, the model's high performance transferred effectively to the unseen validation data, demonstrating good generalization.

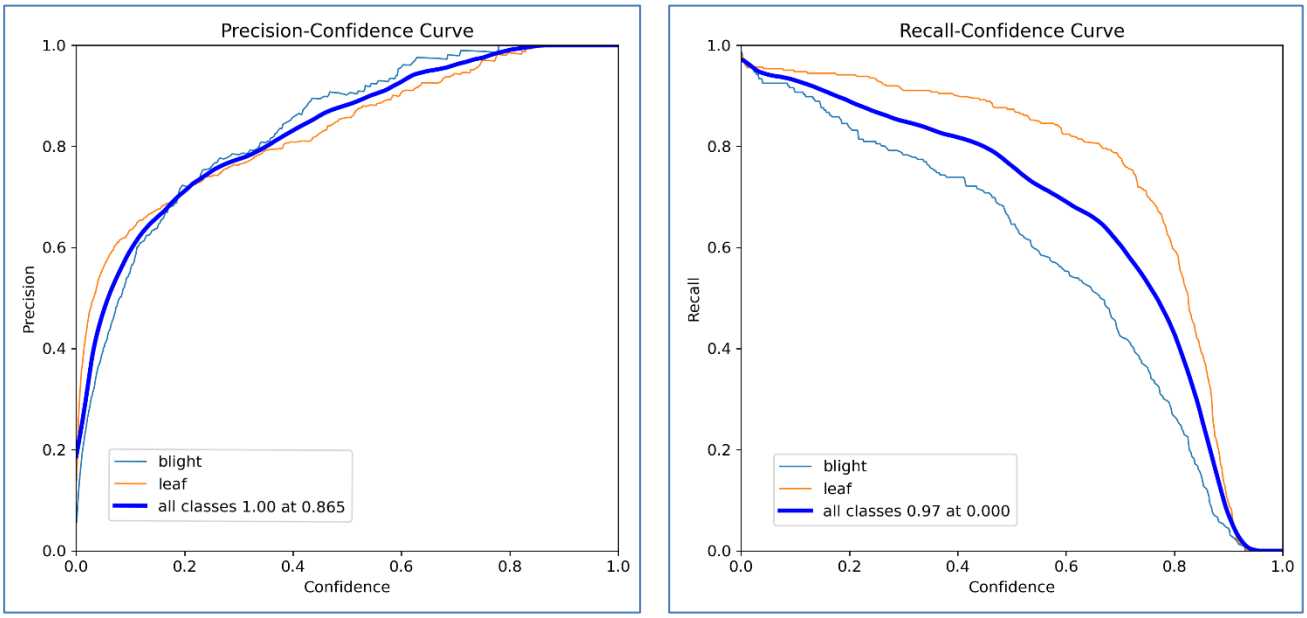
- Overall mAP@0.5: 0.895
- Blight Class mAP@0.5: 0.863 (with a precision of 0.894 and recall of 0.721)
- Leaf Class mAP@0.5: 0.927 (with a precision of 0.813 and recall of 0.895)



(a)



(b)



(c) (d)

Figure 2.17: *Performance evaluation curves for the final model: (a) Precision-Recall, (b) F1-Confidence, (c) Precision-Confidence, and (d) Recall-Confidence.*

Conclusion: In conclusion, the evaluation results confirm the success of the final model configuration. The consistent mAP of 0.895 across both training and validation sets indicates that the model has generalized well and is not overfit. The high precision and recall for both "blight" and "leaf" classes demonstrate its reliability in identifying the targeted features. Combined with the rapid inference speed, these results validate that the chosen YOLOv8n model, trained on the final hybrid dataset, is a robust and efficient solution ready for deployment on the robotic system.

2.3.5 Deployment in ROS:

The trained YOLOv8 model, saved as a .pt file, is deployed into the robotic system using a dedicated ROS 2 node named “blight_detector”. This node is responsible for real-time visual analysis of the environment, and its operational logic is illustrated in the flowchart below (Figure 2.18). Upon initialization, it loads the pre-trained best.pt model weights using the ultralytics library. The core functionality is driven by a timer that periodically triggers a detection cycle. During each cycle, the node captures a frame from a connected webcam. This image is then processed by the YOLO model, which performs

inference to identify objects within the frame, specifically looking for "blight" and "leaf" classes with a confidence threshold of 0.5.

If both classes are detected in the same frame, the node concludes that a blight-infected leaf is present. It then publishes the string "blight" to the “/blight_detection” ROS topic to alert the decision-making strategy. Simultaneously, to provide visual confirmation, the raw image frame is encoded into a base64 string and published to a separate “/blight_image” topic. The node also includes a control mechanism, subscribing to a “/blight_detector/control” topic, which allows the detection process to be paused or resumed via simple string commands, enabling efficient resource management.

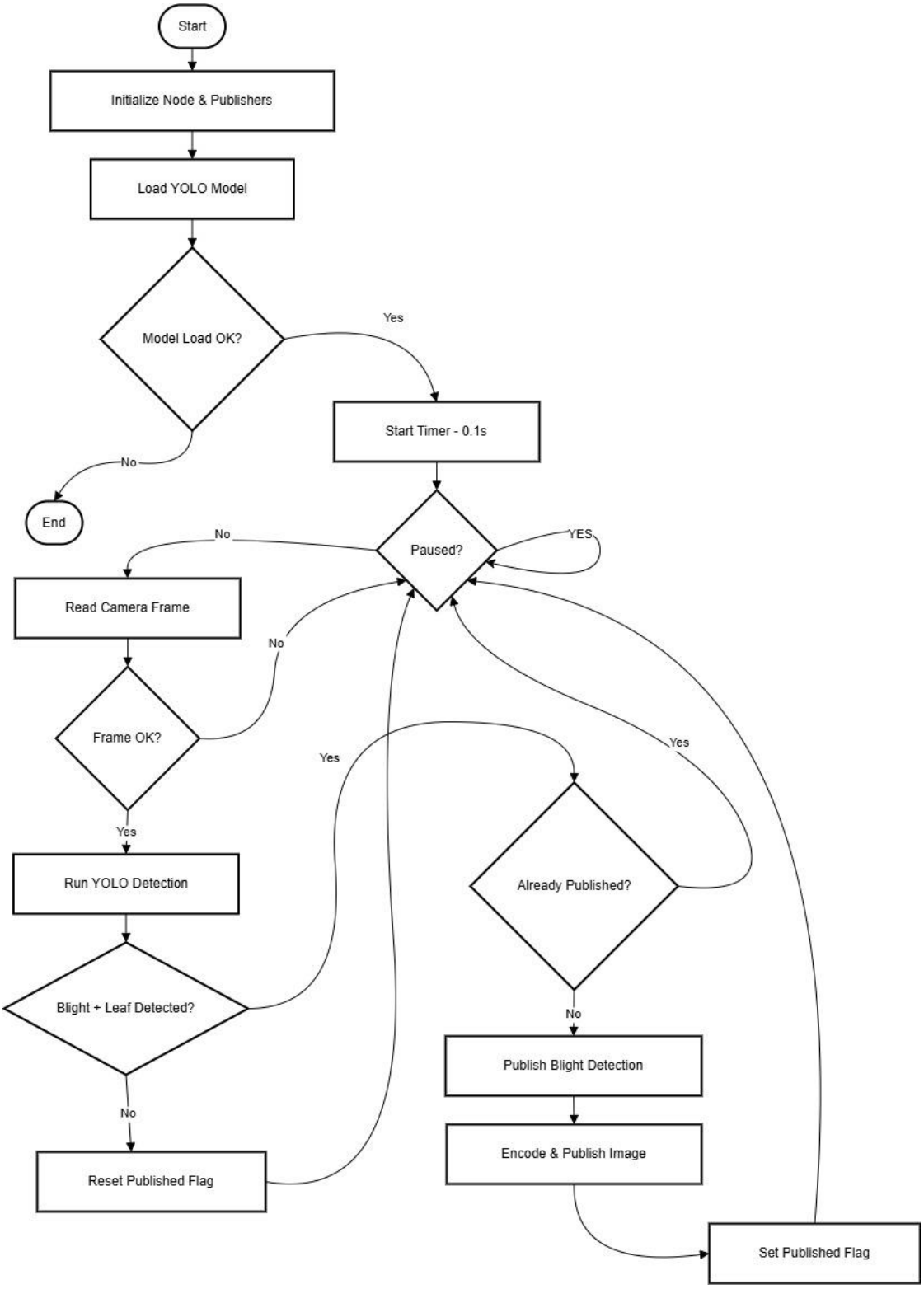


Figure 2.18: *Operational Flowchart of the blight_detector Node.*

2.4 Weather-Based Prediction System

2.4.1 Data Collection

For this project, meteorological data is sourced through two primary weather forecasting APIs: OpenWeatherMap and Open-Meteo. The initial choice, OpenWeatherMap, is a globally utilized platform offering real-time and forecasted weather data. It aggregates information from diverse sources, including weather stations and radar networks, making it a common choice for agricultural applications. However, a comparative evaluation revealed that while its temperature forecasts are relatively accurate, humidity data can exhibit higher variability, which is a significant concern for blight prediction [84].

Due to these limitations, and to improve forecast reliability, the open-source Open-Meteo API was also integrated. Open-Meteo leverages high-resolution meteorological data from reputable sources such as the European Centre for Medium-Range Weather Forecasts (ECMWF), Deutscher Wetterdienst (DWD), and NOAA's Global Forecast System (GFS) [85, 86]. Crucially, preliminary tests indicated that Open-Meteo's humidity forecasts showed a 3% improvement in accuracy over OpenWeatherMap [87]. By integrating both APIs, the system can cross-reference data and leverage the strengths of each, allowing for a more robust and accurate assessment of environmental conditions and disease risk.

2.4.2 Prediction Algorithm

The core of the system is a **Weather-Based Prediction and Spraying Advisory Algorithm**, implemented as a ROS 2 node named "weather_reporter". The operational logic of this node is visually detailed in the flowchart below (see Figure 2.19). This node continuously analyzes meteorological data for the user's specific field location, running on a recurring timer (every 10 minutes). The process begins by acquiring real-time current weather data (air temperature, wind speed) and a detailed 48-hour hourly forecast (temperature, relative humidity, precipitation, wind speed). This chronological

data sequence is then evaluated against two rule-based environmental models. The **Blight Favorability Model** identifies high-risk periods when the air temperature is between 10°C and 25°C and relative humidity exceeds 90%. In parallel, the **Spraying Suitability Model** determines optimal application windows, defined as times when the temperature is between 0°C and 45°C, there is zero precipitation, and wind speed is below 10 m/s. The algorithm systematically iterates through the upcoming 24-hour forecast to find the earliest "Blight-Suitable Time" and the earliest "Spray-Suitable Time." The final output is a structured JSON message containing an actionable forecast, which is published to the "**/weather_status"** ROS 2 topic, providing the current weather status and a prediction for the next high-risk blight period alongside the next optimal window for preemptive spraying.

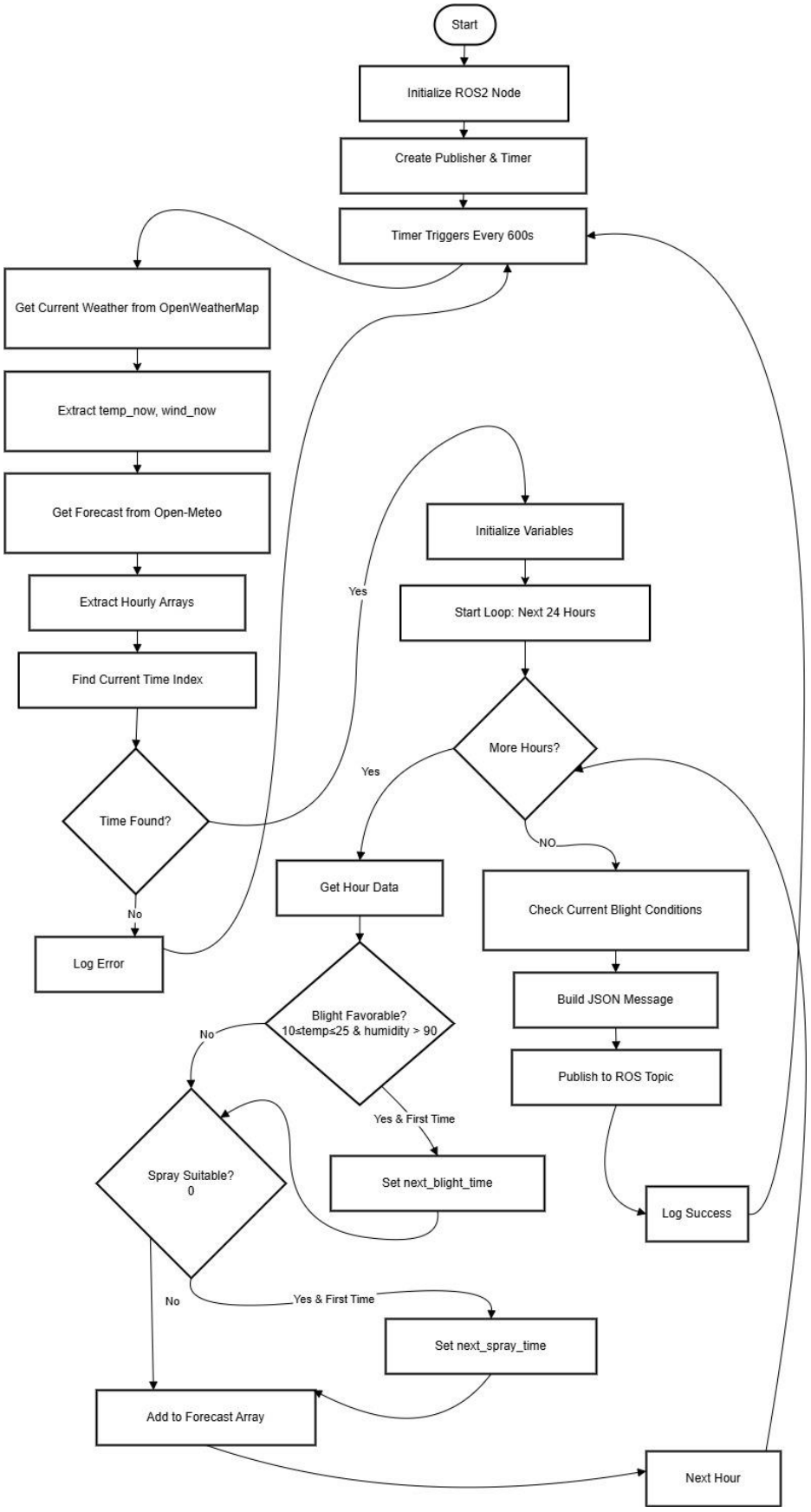


Figure 2.19: *Flowchart of the Weather-Based Prediction and Spraying Advisory Algorithm.*

2.4.3 Integration with the Decision System

The weather prediction algorithm is encapsulated within a dedicated ROS 2 node named “weather_reporter”. This node functions as a standalone module that integrates the prediction logic into the system, as illustrated in the data flow diagram in Figure 2.20. The node does not rely on external topic subscriptions for its primary operation. Instead, its execution is triggered by an internal, periodic timer set to a 600-second (10-minute) interval. At each interval, the node autonomously executes the full data collection and prediction process. The primary output of the “weather_reporter” node is a ROS 2 topic named “/weather_status”. On this topic, it publishes a “std_msgs/String” message containing the analysis results formatted as a JSON object. This JSON string provides a comprehensive summary, including current weather conditions, the predicted time for the next blight-favorable conditions (next_blight_suitable_time), and the next optimal window for spraying (next_spray_suitable_time). This published data is then consumed by the system's primary decision-making node, which subscribes to the “/weather_status” topic. This node is responsible for interpreting the weather analysis and making high-level strategic decisions, such as determining the optimal time to dispatch the robot for a spraying mission.

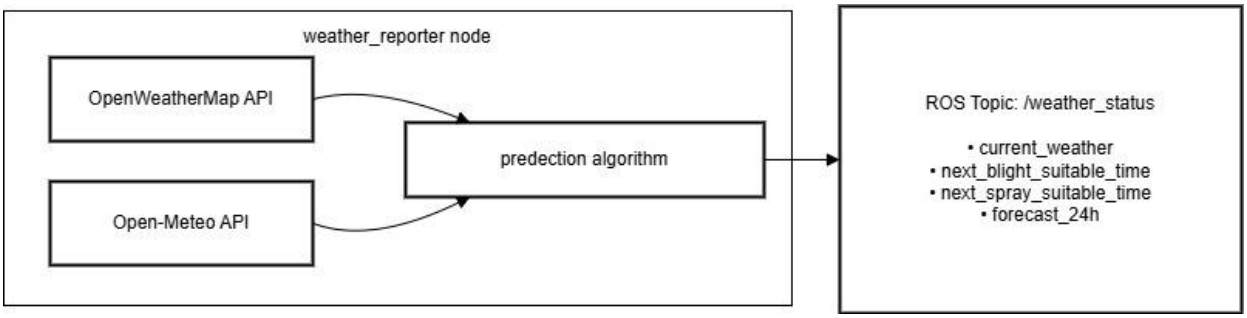


Figure 2.20: *Data Flow Diagram of the weather_reporter Node.*

2.5 Decision-making Strategy and ROS-Based Communication

2.5.1 Node Design and Topics

The system employs a decoupled, publisher-subscriber design where nodes communicate asynchronously through specialized ROS2 topics, creating a resilient and scalable architecture. The flow of information, as illustrated in the system architecture diagram (Figure 2.21), begins with the weather nodes. The "weather_reporter" node publishes comprehensive environmental data and Blight Favorability prediction and Spraying Suitability time to the "/weather_status" topic. The "blight_detector" node, which can be paused and resumed independently via a "/blight_detector/control" topic to save resources, sends out alerts on the "/blight_detection" topic upon identifying an infection.

The central "spray_scheduler" node subscribes to both "/weather_status" and "/blight_detection" to gather the necessary data for its decision-making logic. Once it determines a spray is necessary, it publishes the designated time on the "/spray_schedule" topic. The "spray_decider" node, which orchestrates the physical spraying, is a primary subscriber to this "/spray_schedule" topic.

A critical feedback loop is established through the "/spray_log" topic. When the "spray_decider" initiates a spray, it publishes the start timestamp to "/spray_log". The "spray_scheduler" subscribes to this log, allowing it to confirm that a spray has been executed and prevent scheduling redundant operations. Additional topics facilitate manual control and user interaction. The "/spraying_mode" topic allows switching between manual and automatic operation, the "/spray_cancel" topic allows a scheduled mission to be aborted, and the "/spray_command" topic can be used to trigger an immediate spray. Finally, the "/spray_schedule_notification" topic is used by both the scheduler and the decider to send human-readable status updates to a user interface. This separation of concerns ensures that each node can operate independently, reacting only to the data relevant to its specific task.

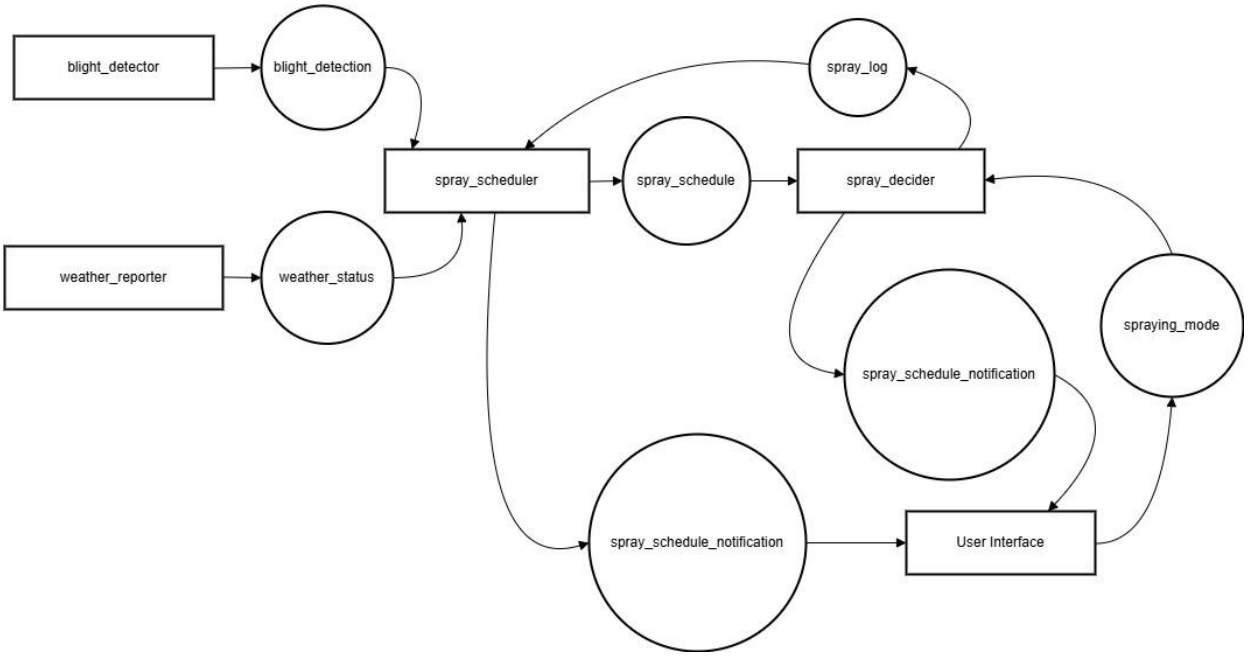


Figure 2.21: *System Architecture and Topic Communication Diagram.*

2.5.2 Decision-making Strategy logic

The decision-making process, visually outlined in the flowchart in (Figure 2.22), is a coordinated sequence across the "spray_scheduler" and "spray_decider" nodes, designed to be both proactive and reactive.

1. **Trigger Identification**: The process begins in the "spray_scheduler" node, which listens for two primary triggers:
    - **Reactive Trigger**: A message on the "/blight_detection" topic indicates a confirmed, active blight infection. This is treated as a high-priority event.
    - **Proactive Trigger**: A message on the "/weather_status" topic containing a "next_blight_suitable_time" indicates that conditions will soon be favorable for a blight outbreak. This allows for preventative action.
2. **Constraint Validation**: Once a trigger is identified, the "spray_scheduler" validates a set of constraints before proposing a mission. It first checks its own

state to ensure a spray mission is not already scheduled. Then, it checks the "/weather_status" data for the "next_spray_suitable_time" to ensure the mission is effective and safe, specifically avoiding scheduling a mission in the current hour to provide lead time. Crucially, it also consults its internal state, which is updated by the "/spray_log", to ensure a spray has not already been performed within the last 7 days. The node also includes a self-correcting timer that automatically cancels proposed schedules if they expire before being executed.

3. **Mission Proposal**: If both a trigger and all constraints are satisfied, the "spray_scheduler" publishes the optimal spray time to the "/spray_schedule" topic. This acts as a formal mission proposal for the rest of the system.

4. **Action and Execution**: The "spray_decider" node receives the mission proposal from "/spray_schedule". Its behavior then depends on its current operational mode:

    - If in **"auto" mode**, it automatically accepts the mission and sets a timer to execute the spray at the scheduled time.

    - If in **"manual" mode**, it notifies the user (via "/spray_schedule_notification") and awaits an "accept" or "decline" message on the "/spray_accept" topic before proceeding.

5. **Closing the Loop**: Once a spray mission is executed (either from a schedule or an immediate command), the "spray_decider" begins spraying for a fixed duration of 60 seconds. At the start of this period, it publishes a timestamp to "/spray_log". This message is received by the "spray_scheduler", which updates its "last_spray_time" state, successfully closing the decision loop and preventing redundant actions.

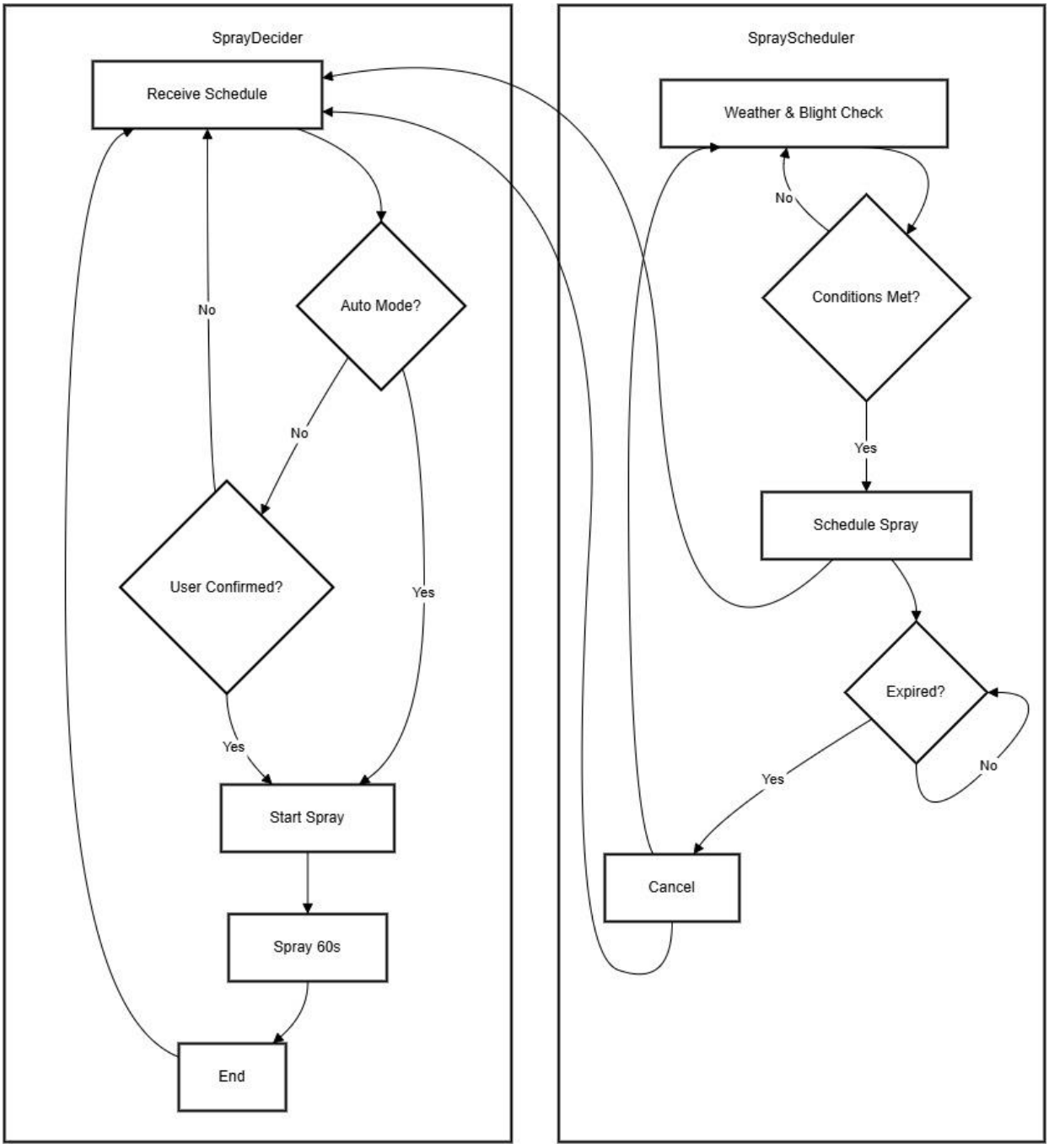


Figure 2.22: *Flowchart of the Decision-Making Logic.*

Chapter 3 System Integration

3.1 Overview of System Integration

This chapter addresses the critical phase of **System Integration**, where the system's theoretical and algorithmic foundations are translated into a functional, real-world prototype. The high-level architecture of this integrated system, illustrated in (Figure 3.1), involves integrating the AI-driven software, which handles perception and decision-making, with a custom-built physical robotic body platform designed for mobility and durability in agricultural environments. The core hardware components include the mobile robot chassis, a precision spraying mechanism for targeted treatment, and a web-based user interface that allows the farmer to command, monitor, and manage the entire system remotely. The Robot Operating System (ROS) serves as the central framework, providing the necessary communication and synchronization between all software modules and hardware components to create a cohesive, end-to-end crop management solution.

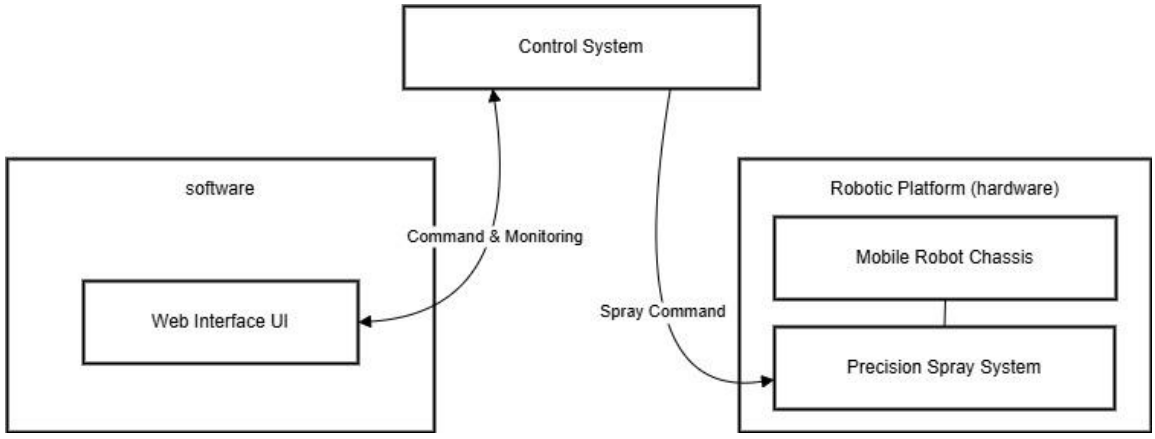


Figure 3.1: *High-level system architecture, illustrating the interaction between the control system, the physical robotic platform, and the web interface.*

3.2 Robot Mechanism Design

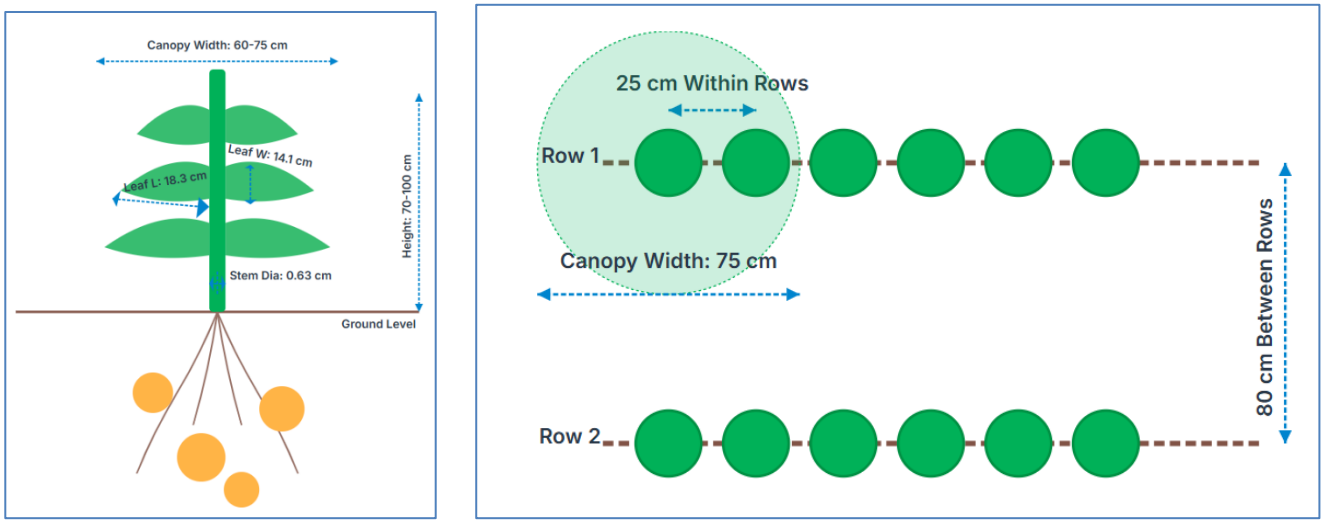
3.2.1 Field and Plant Analysis

Understanding the physical characteristics of potato plants is crucial for designing effective robotic systems, particularly for detecting fungal infections and applying fungicides precisely. This study focuses on the Spunta potato variety, which dominates cultivation in Algeria, accounting for approximately 40% of the country's potato production (5). Therefore, the robot's design and operational parameters are tailored specifically to this variety.

a. *Study of potato plant dimensions and spacing*

Potato plant dimensions exhibit considerable variation influenced by environmental factors, cultivation techniques, and genetic characteristics. Under controlled aeroponic conditions with optimized nutrients and environmental parameters, Spunta and other cultivars achieve heights of 150–180 cm, averaging approximately 93 cm at 63 days after transplanting (DAT) [88–90]. In open-field environments, Spunta typically attains 60–100 cm, though exceptional cases under ideal irrigation and fertility reach 120–130 cm in varieties like Belete and Gudanie [88–90]. Within Algerian agricultural settings, Spunta demonstrates vigorous growth with maximum heights of 70–100 cm (see figure 3.2.a), contingent upon seasonal conditions and field management practices [66]. Regarding lateral development, Spunta exhibits canopy widths of 60–75 cm. This growth pattern correlates directly with standard Algerian row spacing configurations of approximately 80 cm between rows and 20–30 cm within rows (see figure 3.2.b), facilitating near-complete canopy coverage under high-fertility irrigation [66, 91]. Stem structural properties include robustness, with seedling stems measuring 7–12 cm featuring 5–8 mature leaves, while main stems display diameters of approximately 6.3 mm under experimental conditions (see figure 3.2.a), supporting upright growth and canopy expansion [88, 92]. Leaf morphology parameters are similarly significant, with the fourth leaf of Spunta plants averaging 18.3 cm in length and 14.1 cm in width during controlled studies (see figure

3.2.a), establishing critical baselines for precision agriculture applications [88, 92].



(a) **(b)**

Figure 3.2: *Diagram of potato plant morphological dimensions (a) and its standard field planting geometry (b)*

b. *Field Structure and Navigation Constraints*

The adopted row spacing of 80 cm and within-row spacing of 25 cm, combined with Spunta's typical canopy width of 75 cm, creates a tightly structured field environment. This configuration results in minimal inter-row clearance (effectively ~5 cm per side) and dense intra-row plant placement, necessitating precise robot control to navigate between rows without damaging adjacent canopies. However, the moderate maximum plant height of 90 cm facilitates over-canopy operation, enhancing the robot's field of vision and accessibility for both disease detection and targeted spraying.

Critical operational parameters for the robotic platform were established through rigorous analysis of Spunta potato cultivation practices in Algeria. The foundational measurements include:

- Row spacing: Standardized at 80 cm based on prevalent field configurations.
- Canopy width: Documented as 75 cm, representing typical lateral spread.
- Within-row spacing: Defined as 25 cm, reflecting optimal planting density.
- Maximum plant height: Set at 90 cm, accounting for growth variability under field conditions.
- Operational clearance: Derived as 5 cm per side from the difference between row spacing (80 cm) and canopy width (75 cm).

This constrained spatial environment necessitates an adaptive width capability in the robotic design. This core functional requirement ensures safe traversal capabilities within inter-row corridors while accommodating natural biological variability in canopy development.

3.2.2 Robot Mechanical Design

The robotic system prioritizes three critical design objectives to address field operational challenges:

- **Adaptability:** Essential for accommodating varied field geometries across Algerian agricultural landscapes, where spatial layouts exhibit substantial variation between plots.
- **Stability:** Must be maintained throughout navigation to ensure consistent platform orientation, enabling accurate imaging and targeted spray deployment.
- **Accessibility:** Requires comprehensive coverage of all plant structures, —particularly lower foliage where late blight infections typically initiate —to support effective disease intervention.

These goals collectively enable reliable performance in unstructured field environments.

3.2.3 Robot Structure Overview

To address the aforementioned constraints, several versions have been developed. Each version aimed to systematically resolve the shortcomings of the previous design, focusing on enhancing stability, mobility, and overall robustness for agricultural fieldwork.

a. *Version V1.0:*

The first design version introduced a table-like structure serving as the robot's main body (see Figure 3.3). Four horizontal beams extend outward from the central platform; each connected via pivot joints to both the central platform and a vertical pillar. At the bottom end of each pillar is a single fixed wheel, enabling movement. Notably, the entire leg assembly relies on the rotation of the vertical pillars to change direction, eliminating the need for a traditional steering mechanism. This configuration aimed to give the robot width adaptability, ideal for navigating between rows of trees or crops with varying spacing. The symmetrical layout allows the robot to straddle over vegetation. However, deeper inspection of the model reveals structural and functional issues. The long horizontal arms are unsupported and prone to flexing under load, especially near the pivot joints. Additionally, the concentration of forces on the central plate and minimal bracing reduce overall stability. The reliance on pillar rotation for steering increases mechanical complexity and power consumption. These design weaknesses made the system fragile, inefficient, and unsuitable for uneven terrain or sustained field use, leading to the development of an improved version.

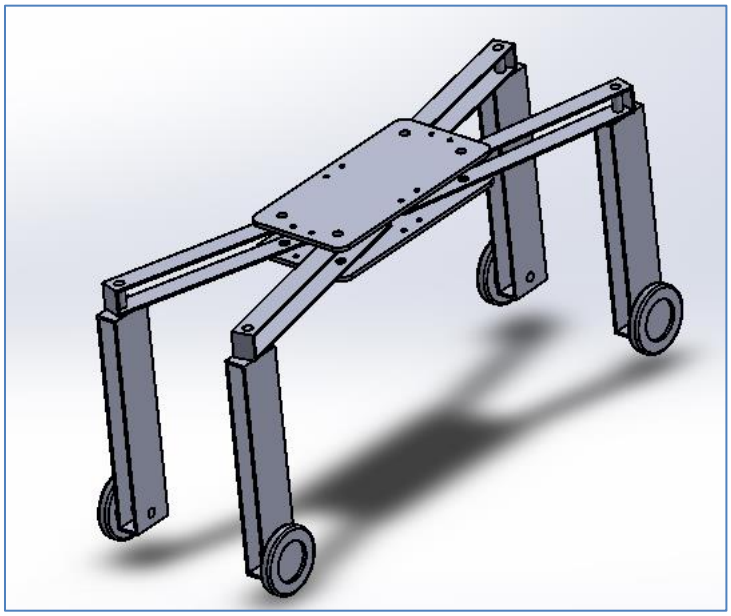


Figure 3.3: *Robot Structure V1.0*

b. *Version V2.0:*

In this second version, significant improvements were introduced to address the structural weaknesses of V1.0 (see Figure 3.4). The robot's legs are now inclined outward, rather than mounted vertically. This angled design enhances overall mechanical strength and allows the robot to better distribute its weight, improving both stability and durability during motion.

The main platform has been widened, reducing the center of a gravity and increasing ground clearance between the wheels. This makes the structure more stable on uneven surfaces and less likely to tip during operation.

A major upgrade was made in the steering and mobility mechanism: the bulky vertical pillars were replaced with compact, independent wheel mounts, each capable of rotation.

These small wheel holders offer higher control precision, especially during tight turns or when navigating irregular paths between crops. The wheels are a caster type, which allows for omni-directional movement. This change also reduces mechanical complexity, weight, and energy consumption compared to the previous pillar-based steering system.

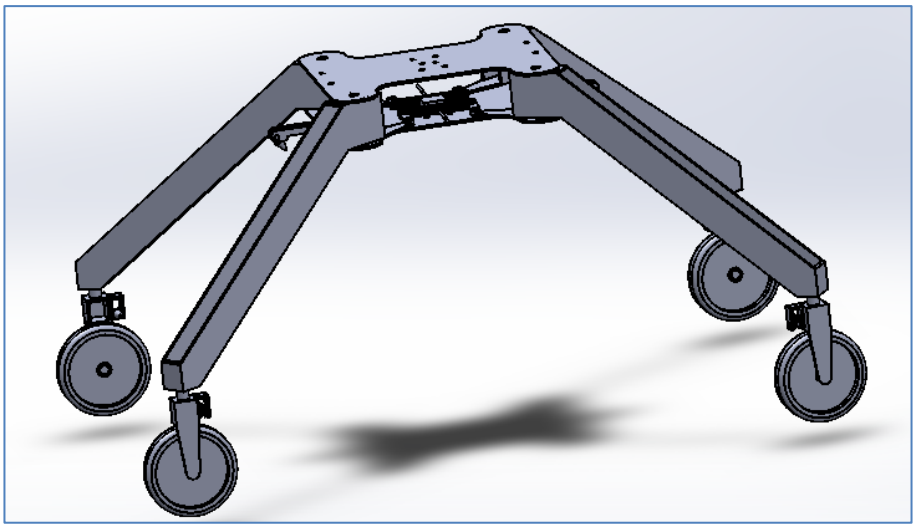


Figure 3.4: *Robot Structure V2.0*

c. *Version V2.1:*

This version focuses on compactness, balance, and terrain adaptability (see Figure 3.5). The most visible structural change is the inward angling of the legs, which significantly reduces the overall footprint of the robot. This new geometry not only shortens the base length but also centers the mass beneath the main platform, enhancing stability, especially when operating on uneven ground.

To support better maneuverability in agricultural environments, larger and wider wheels were introduced. These wheels are more suited for soft soil, offering better grip, smoother

rolling, and reduced chances of sinking or slipping. Their size also increases ground clearance slightly, giving the robot a better ability to pass over small obstacles like roots or rocks.

The leg configuration now adopts a more vertical posture with slight inward inclination, which reduces torque on the chassis and contributes to a more compact, lightweight design.

Internally, the connections between the legs and the chassis have been simplified, removing redundant supports and making the assembly easier to manufacture and maintain. This version is practical for today's work, and its simple design makes it easy to upgrade with better sensors or wheels in the future.

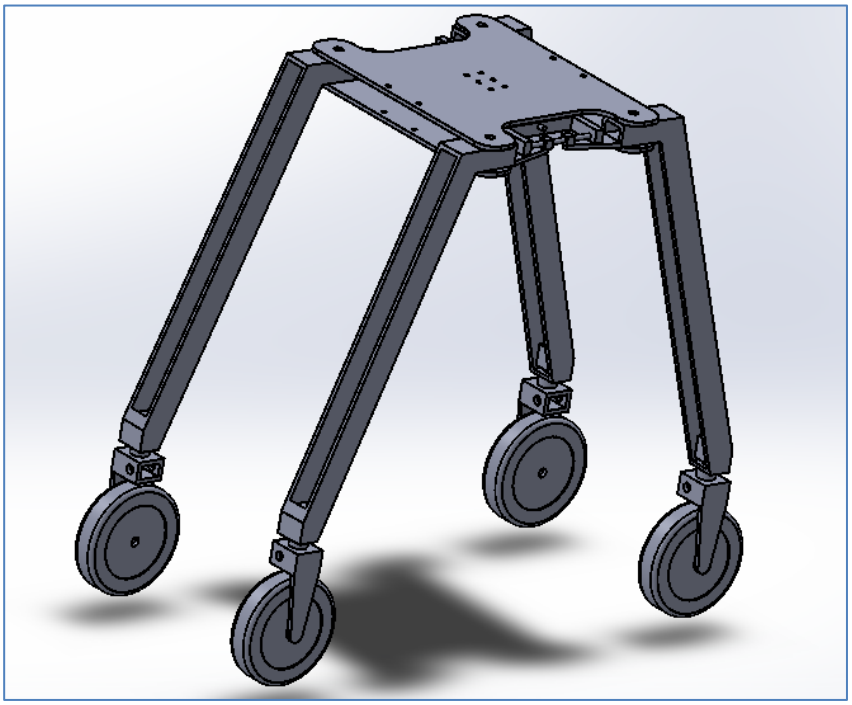


Figure 3.5: *Robot Structure V2.1*

3.2.4 Adjustable Wheelbase and Structural Layout

The robot features a dynamically adjustable wheelbase that allows it to adapt to varying crop row widths, enhancing its versatility across diverse field geometries. The structural frame is constructed from lightweight aluminum, offering a balance of strength, durability, and ease of manufacturing. The platform is mounted on four wheeled legs connected through a scissor-like mechanism, forming a stable elevated chassis. A single central motor drives a gearbox with one perpendicular input and two lateral outputs; each connected to a pair of legs on the left and right sides. These outputs actuate lead screw–nut assemblies that drive a sliding mechanism, enabling the legs to extend or retract symmetrically (see Figure 3.6).

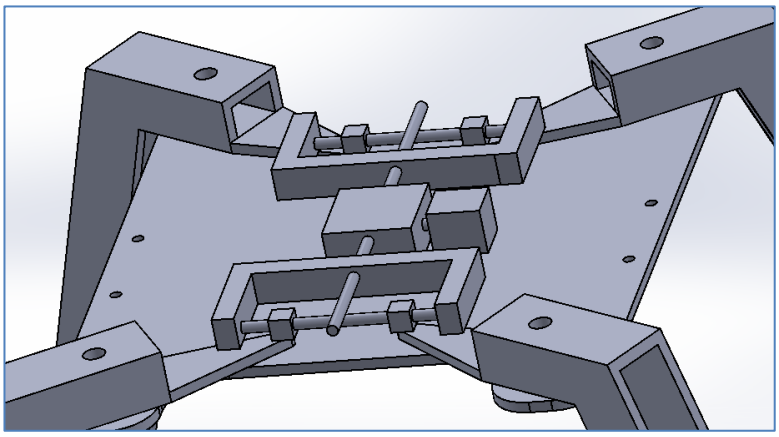
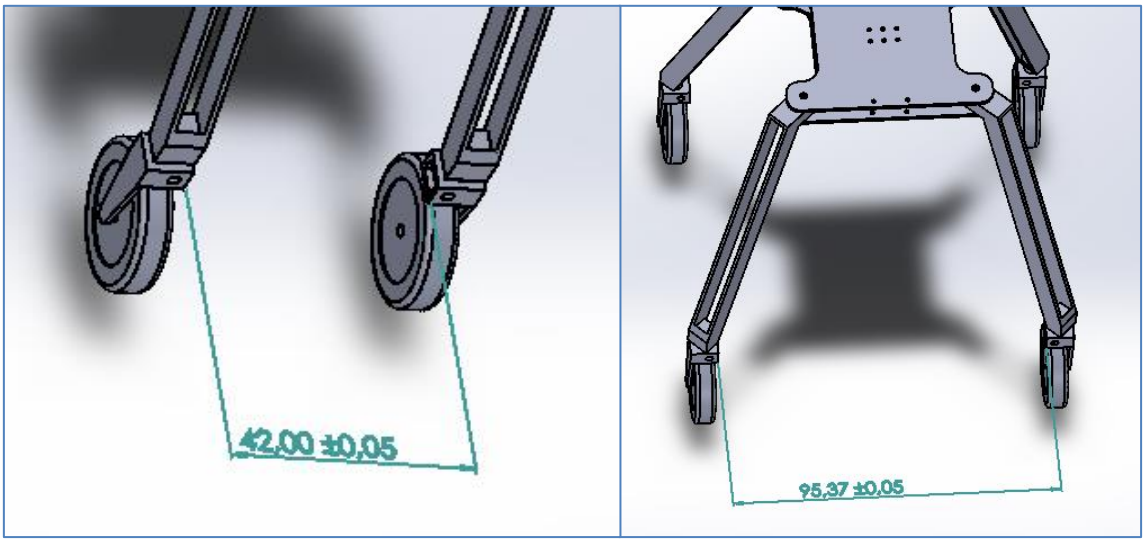


Figure 3.6: *Drive mechanism for wheelbase adjustment*

This mechanism effectively translates the linear push/pull motion from the lead screws into a lever-like action on the legs. This allows for a smooth and coordinated adjustment of the wheelbase width from 42 cm to 95 cm, without compromising structural integrity (see Figure 3.7).



(a) (b)

Figure 3.7: *Adjustable wheelbase range, showing (a) minimum width and (b) maximum width.*

The height from the ground to the lower surface of the platform is fixed at 112 cm, ensuring consistent clearance above the crop canopy, while the legs are angled outward at 110° to enhance lateral stability (see Figure 3.8).

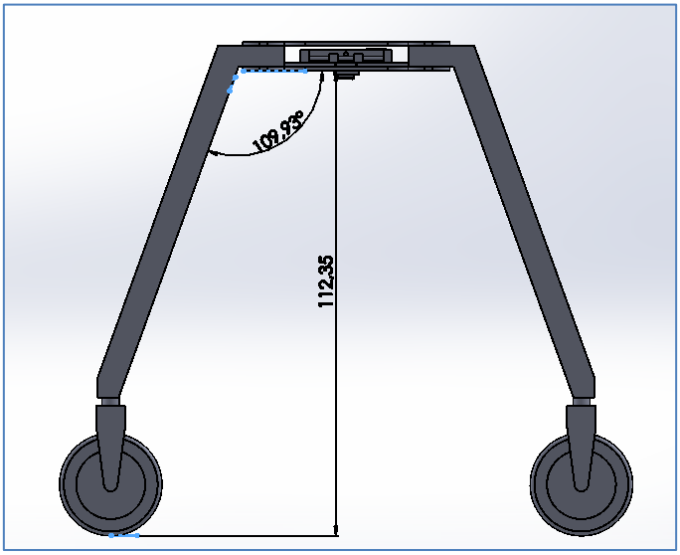
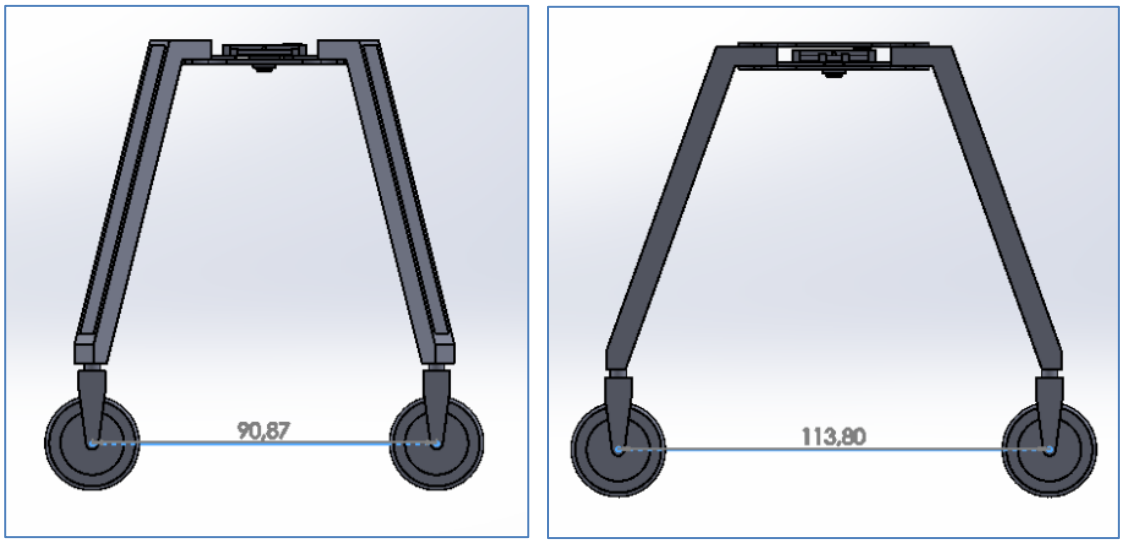


Figure 3.8: *Side profile view showing the robot's height and leg angle.*

The distance between the front and rear wheels ranges from 93 cm to 113 cm, depending on the adjusted width, due to the direct relationship between leg extension and horizontal spread (see Figure 3.9). This dynamic structural adaptability allows the robot to maintain optimal alignment over crop rows, ensuring both precise navigation and unobstructed operation of its sensing and spraying systems.



(a) (b)

Figure 3.9: *Front-to-rear wheel distance variation, showing (a) the minimum distance at maximum width and (b) the maximum distance at minimum width.*

3.2.5 Camera Mounting and Vision Coverage

The camera is centrally mounted on the underside of the robot's main platform, oriented directly downward (see Figure 3.10). Positioned at a fixed height of 112 cm above the ground, this placement offers a wide, unobstructed field of view across the crop canopy. The elevated and centralized configuration is designed to capture high-resolution images focused on the upper leaf surfaces, which are most indicative of plant health.

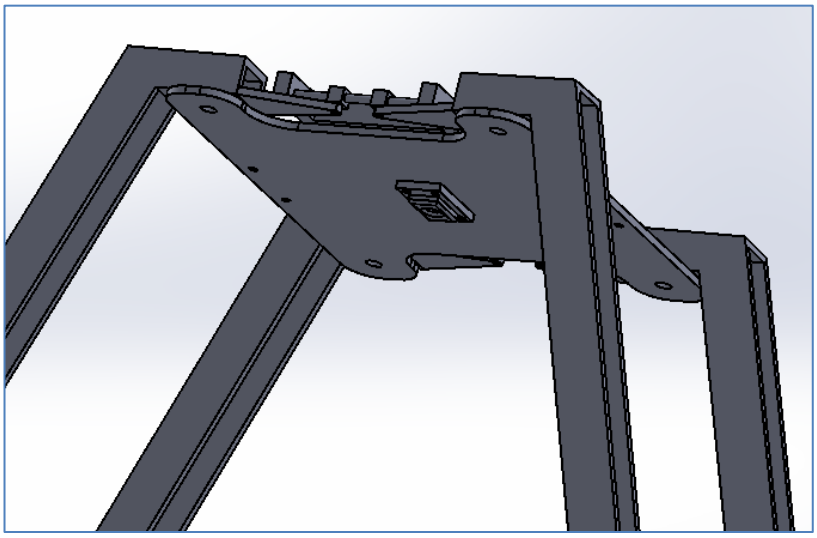


Figure 3.10: *Underside view of the robot platform showing the central mounting position of the camera.*

By aligning the camera with the geometric center of the robot's base, uniform image coverage is achieved as the robot moves along crop rows. This strategic positioning ensures consistent data acquisition, minimizes occlusions, and enhances the accuracy of the vision system. Overall, this setup significantly improves the system's ability to detect, localize, and monitor plant health conditions with precision.

3.2.6 Description of the 3D Model in SolidWorks

The complete mechanical structure of the robot was designed as a detailed 3D model in SolidWorks to ensure precision in manufacturing and assembly. The design focuses on a modular, symmetrical, and robust frame capable of navigating agricultural environments while supporting the necessary electronic and mechanical subsystems. Different views of the model, including an isometric perspective of the final assembly (Figure 3.11), an exploded view (Figure 3.13), and standard orthographic projections (Figure 3.12), provide a comprehensive understanding of the robot's construction.

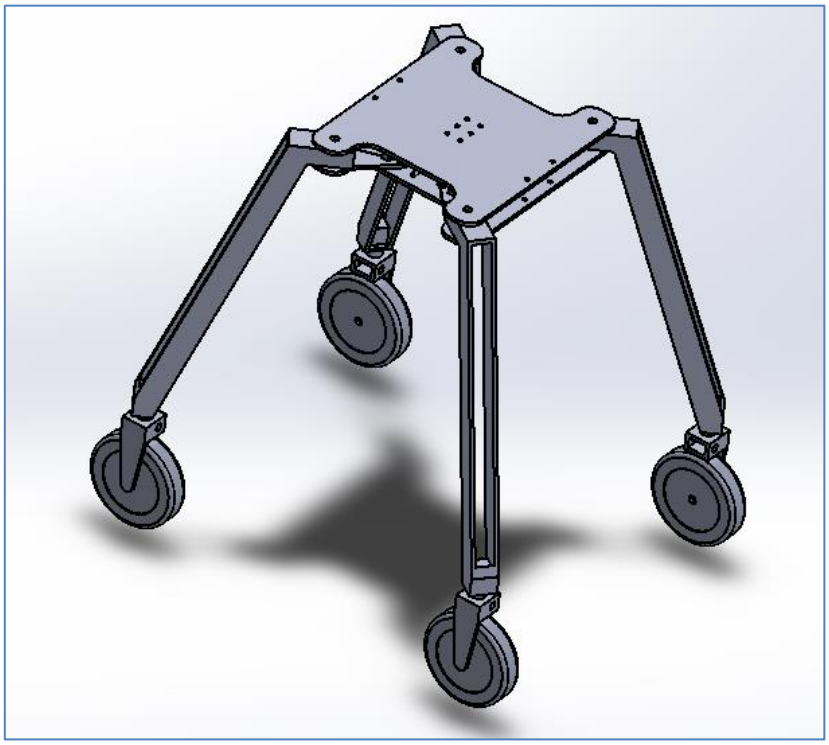


Figure 3.11: *Isometric view of the assembled V2.1 robot model.*

d. *Structural Assembly:*

The robot's chassis is built around a central platform (BASE1), which acts as the main structural hub. The symmetrical nature of the design is clearly visible in the top-down orthographic view (Figure 3.12c), which shows how the leg assemblies are perfectly mirrored. This flat, rectangular component is designed with specific cutouts and mounting holes to securely attach the leg assemblies and the wheelbase adjustment mechanism. The underside of the platform also features a dedicated, centralized mount (CAM in Figure 3.13) designed to hold the vision system's camera in a fixed, downward-facing orientation. The four support legs are two-part assemblies (LEG BAS1 and LEG1) that connect to the underside of the main platform. The side and front profiles (Figure 3.12a and 3.12b) best illustrate the aggressive outward angle of the legs, which creates a very stable, trapezoidal footprint crucial for operating on uneven terrain. At the end of each leg, a simple, durable wheel (WHEEL1) is mounted, allowing for mobility.

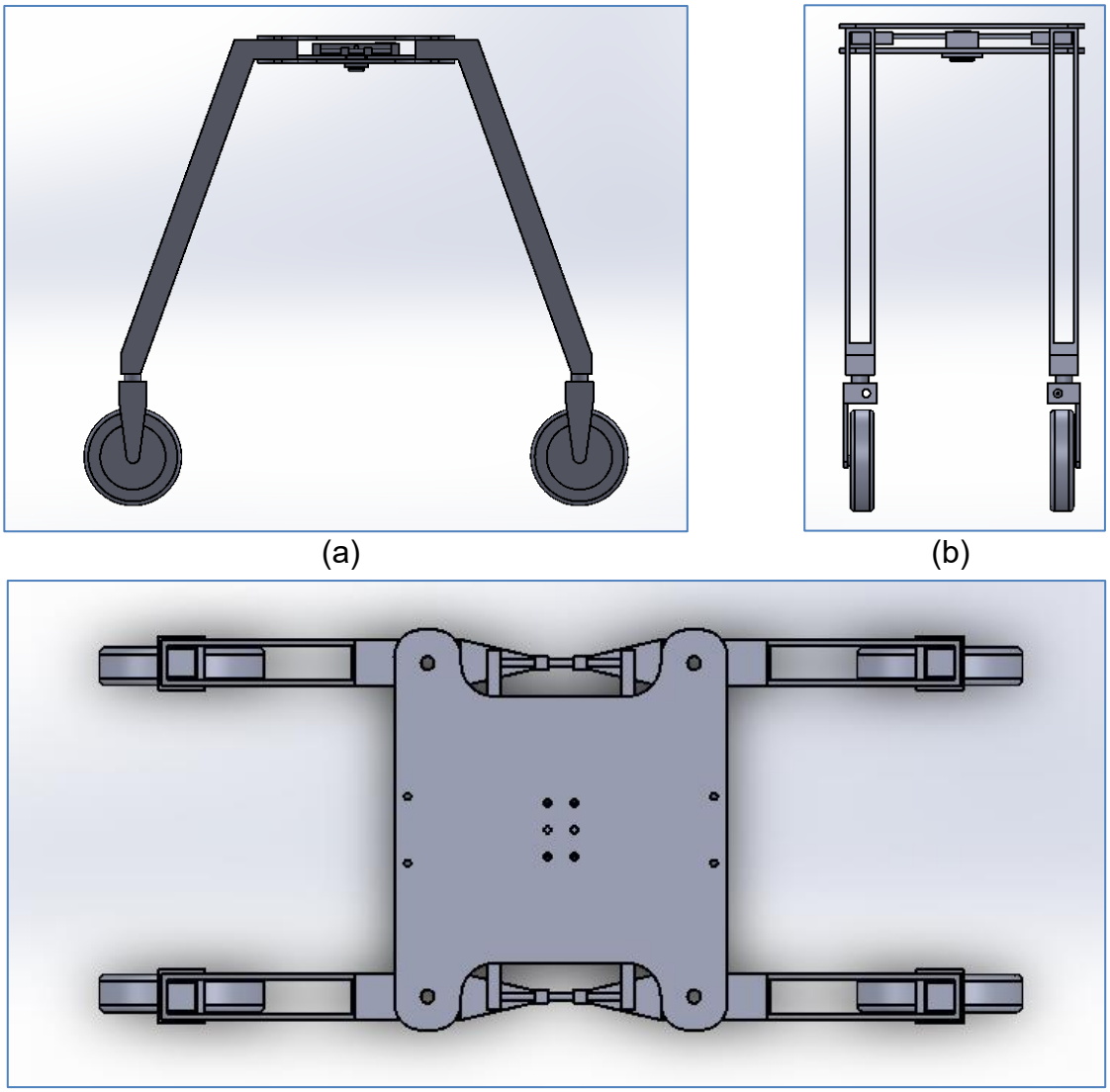


Figure 3.12: *Orthographic views of the robot model: (a) Side view, (b) Front view, and (c) Top view.*

e. *Wheelbase Adjustment Mechanism:*

A key feature of the design is the dynamic wheelbase adjustment system, which is clearly visualized in the 3D model. This mechanism is driven by a central gearbox (gear_box). The gearbox actuates a sliding lead-screw system. The exploded view (Figure 3.13) provides a clear look at this system's individual components, showing precisely how the parts labeled Part8G and Part69 interlock to form the sliding housing for the lead-screw assembly. This assembly converts the rotational motion from a single motor into a synchronized linear force, pushing the leg assemblies outward or pulling them inward. This design allows the robot's width to be smoothly adjusted to match varying crop row spacing.

The exploded view of the model (Figure 3.13) highlights the modularity of the design, showing how each component fits together. This detailed digital blueprint was essential for verifying component compatibility and for guiding the subsequent manufacturing and assembly processes.

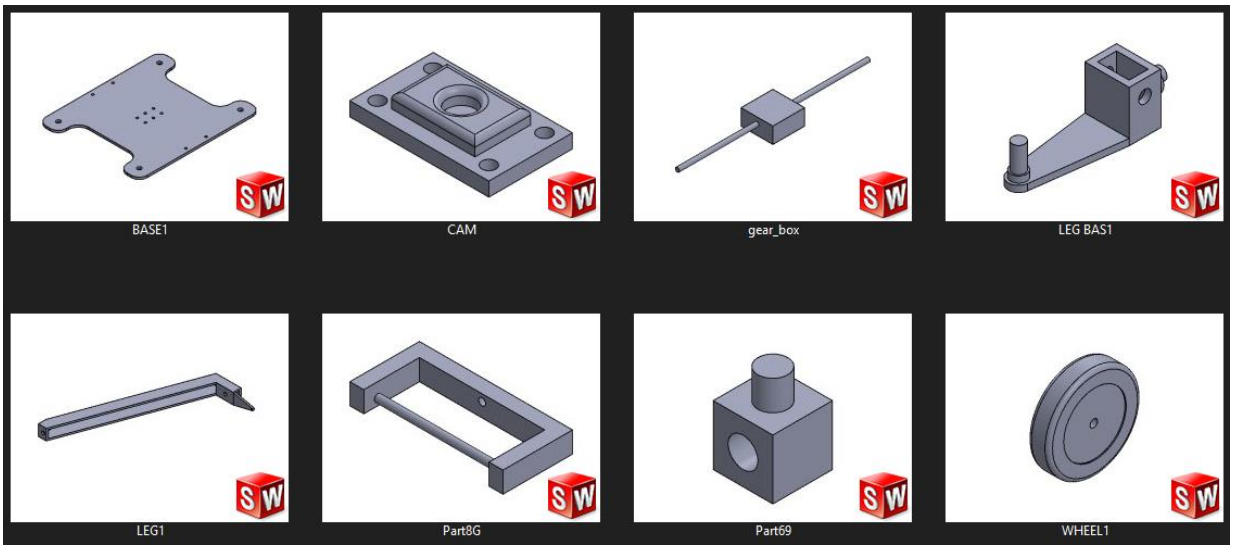


Figure 3.13: *Exploded view of the robot's 3D model showing the main components.*

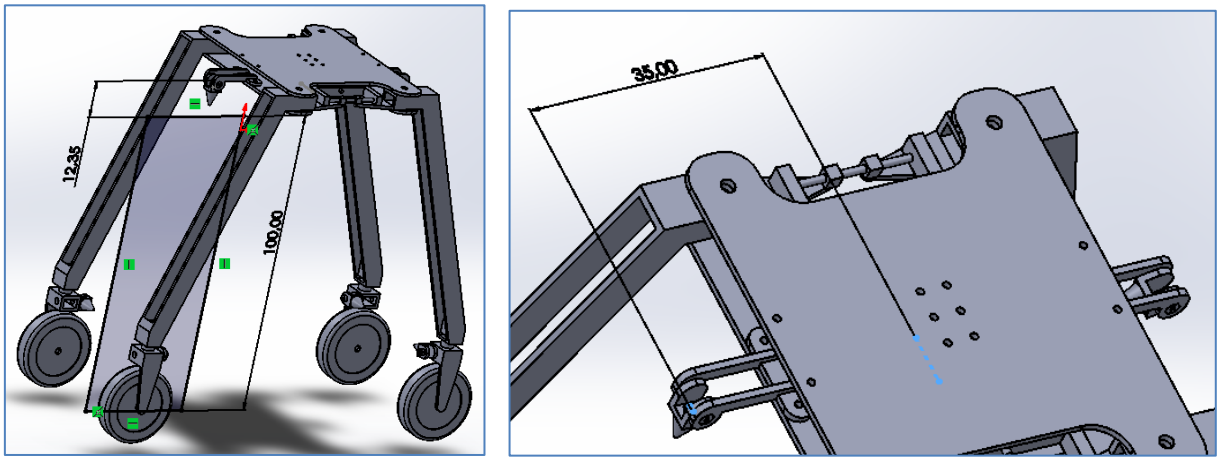
3.3 Precision Spraying System

3.3.1 Spraying Geometry and Environmental Assumptions

To design a robust spraying system, the spatial and agronomic parameters of the field and crop must be defined. The crop of focus is the Spunta potato variety, which typically grows to a height of 90 cm with an average canopy width of 75 cm. The spraying robot features two categories of nozzle placements: top-mounted and leg-mounted nozzles, with their key dimensions shown in the CAD models in (Figure 3.14) and (Figure 3.15).

- **Top nozzles** are positioned 35 cm laterally from the robot's centerline (Figure 3.14b) and are elevated 12 cm above the plant canopy (Figure 3.15a).
- **Leg nozzles** are located 10 cm away from the crop rows and elevated 29 cm from the ground (Figure 3.15).

The objective is to provide uniform coverage of the potato plants, ensuring full foliar and vertical surface exposure while avoiding blind spots. To that end, the target spray area includes a 1.5m width at the top of the canopy (80cm per nozzle with overlap of 10cm) and a 1.5m (80cm per nozzle with overlap of 10cm) lateral spread at ground level, along with complete coverage of the 90 cm plant height.



(a) (b)

Figure 3.14: *CAD model illustrating the placement of the top-mounted nozzles, showing (a) the elevation above the plant canopy and (b) the lateral distance from the robot's centerline.*

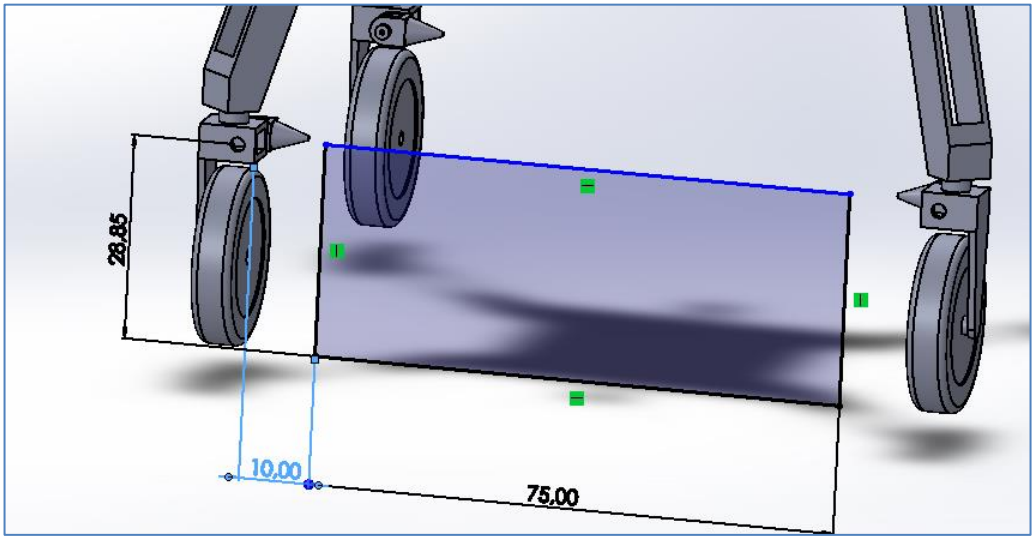


Figure 3.15: *CAD diagram detailing the vertical and horizontal placement of the leg-mounted nozzles relative to the ground and crop row.*

3.3.2 Nozzle Orientation and Spray Angle Requirements

a. *Top Nozzles*

For the top nozzles, which are aimed directly downwards, there is no need for inclination. Each top nozzle must cover a width of 80 cm (40 cm to either side) from an elevation of 12 cm. Using the geometric analysis shown in (Figure 3.16), we can determine the required spray angle.

- **Calculation:** Let a be the half-angle of the spray cone.

$$\tan(a) = \frac{opposite}{adjacent} = \frac{40\ cm}{12\ cm}\ a = \arctan\left(\frac{40}{12}\right) \approx 73.3\circ$$

Required Total Spray $\boldsymbol{Angle = 2 \times a \approx 146.6\circ}$

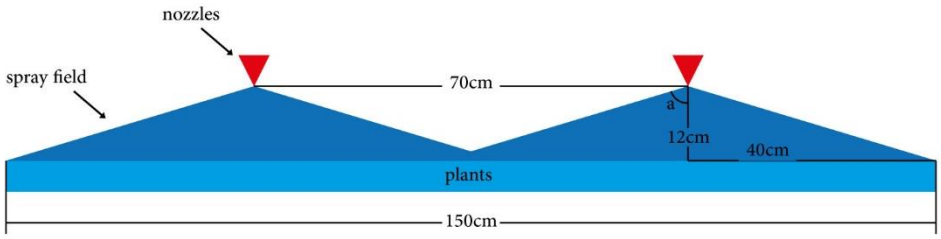


Figure 3.16: *Conceptual diagram of the overlapping spray fields from the top-mounted nozzles.*

b. *Leg-Mounted Nozzles*

The leg-mounted nozzles must be precisely oriented to cover both the full vertical height of the plant and the required lateral width on the ground. This requires calculating both a spray angle and an inclination angle for the vertical and horizontal planes.

- **Vertical Orientation (Covering Plant Height):** As illustrated in (Figure 3.17a), the nozzles are positioned at a height of 29 cm and must cover the plant up to its full height of 90 cm. The nozzle is 10 cm away from the plant row.

Calculation: Vertical distance to cover $\boldsymbol{upwards} = \mathbf{90}\ \boldsymbol{cm} - \mathbf{29}\ \boldsymbol{cm} = \mathbf{61}\ \boldsymbol{cm}$

Let $\boldsymbol{b_1}$ be the angle to spray downwards and $\boldsymbol{b_2}$ be the angle to spray upwards, relative to the horizontal plane.

$$\boldsymbol{b_1} = \mathbf{arctan}\left(\frac{\boldsymbol{downward_{distance}}}{\boldsymbol{lateral_{distance}}}\right) = \mathbf{arctan}\left(\frac{\mathbf{29}}{\mathbf{10}}\right) \approx \mathbf{71.0}^{\circ}$$

$$\boldsymbol{b_2} = \mathbf{arctan}\left(\frac{\boldsymbol{upward_{distance}}}{\boldsymbol{lateral_{distance}}}\right) = \mathbf{arctan}\left(\frac{\mathbf{61}}{\mathbf{10}}\right) \approx \mathbf{80.7}^{\circ}$$

Required Vertical Spray Angle $(\boldsymbol{b}) = \boldsymbol{b_1} + \boldsymbol{b_2} \approx \mathbf{151.7}^{\circ}$

Required Vertical Inclination $(\boldsymbol{Iv}) = \boldsymbol{b_2} - \boldsymbol{b_1} \approx \mathbf{9.7}^{\circ}$ (angled upwards)

- **Horizontal Orientation (Covering Ground Width):** As illustrated in Figure 3.17b, each leg nozzle must cover a horizontal width of 80 cm from a lateral distance of 10 cm. The spray is asymmetric, covering 27 cm to front and 53 cm to the back depending on the front-to-rear wheel distance at maximum width that is 90cm (see section 3.2.4), so

  $$\frac{\mathbf{90}\boldsymbol{cm}}{\mathbf{2}} = \mathbf{45}$$

  and we add the overlap

  $$\mathbf{45}\boldsymbol{cm} + \mathbf{10}\ \boldsymbol{cm} = \mathbf{55}\boldsymbol{cm}$$

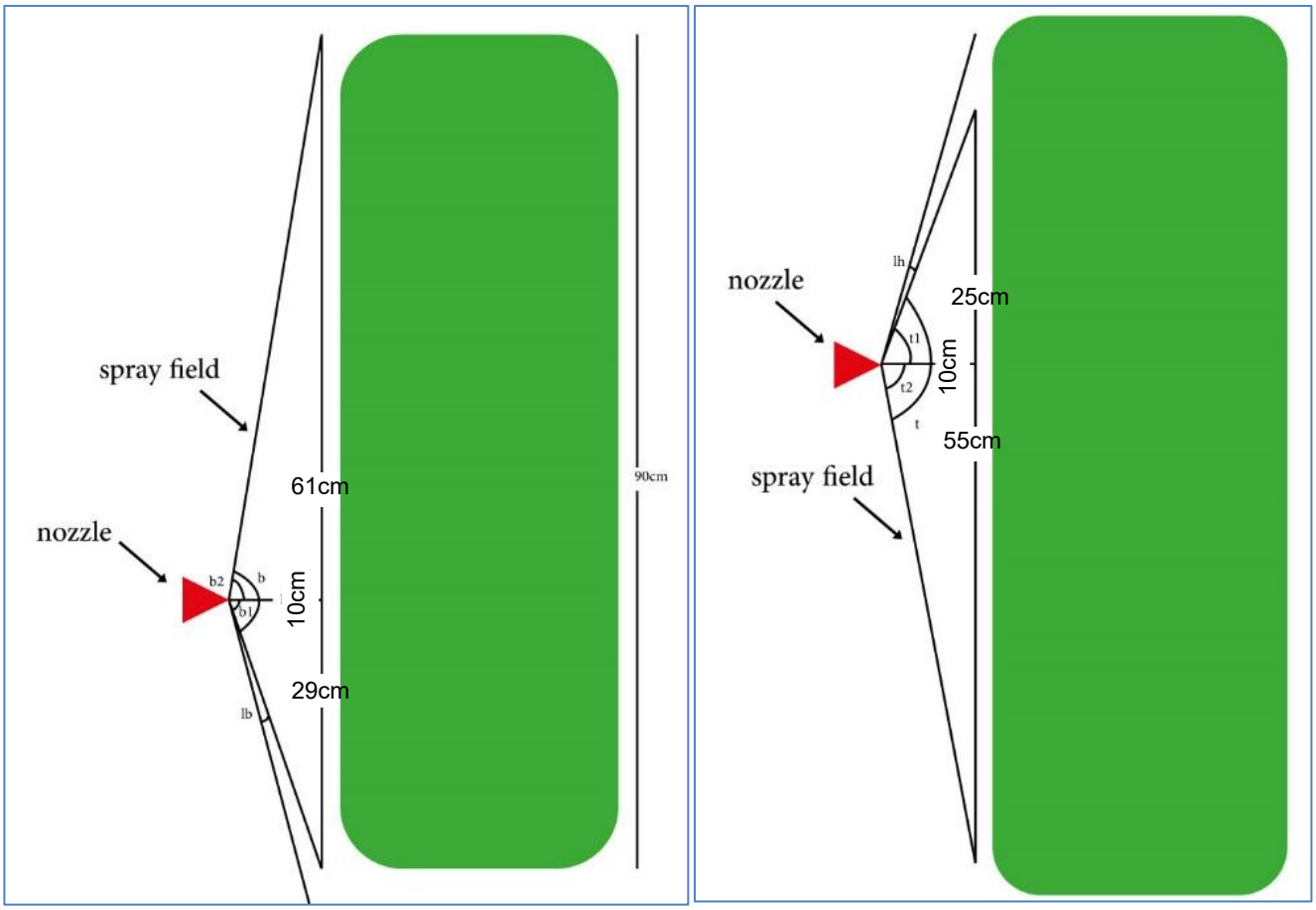
Calculation: Let $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$ be the angles required to cover the two horizontal sections.

$$\boldsymbol{t1} = \boldsymbol{arctan}\left(\frac{\mathbf{25}}{\mathbf{10}}\right) \approx \mathbf{68.2}^{\circ}$$

$$\boldsymbol{t2} = \boldsymbol{arctan}(\frac{\mathbf{55}}{\mathbf{10}}) \approx \mathbf{79.7}\circ$$

Required Horizontal Spray Angle $(\boldsymbol{t}) = \boldsymbol{t_1} + \boldsymbol{t_2} \approx \mathbf{147.9}^{\circ}$

Required Horizontal Inclination $(\boldsymbol{Ih}) = \boldsymbol{t_2} - \boldsymbol{t_1} \approx \mathbf{11.5}^{\circ}$



(a) (b)

Figure 3.17: *Geometric analysis for determining the (a) inclinations from side view and (b) spray angles from top view, for the leg-mounted nozzles.*

3.3.3 Nozzle Type and Material Selection

Given the robot's stationary spraying behavior and the need for full area saturation, full cone nozzles were deemed most appropriate. Full cone nozzles provide uniform coverage across a circular footprint, making them ideal for pesticide and fungicide applications on dense canopies. The next consideration is material compatibility. Curzate M contains mancozeb, a chemical known to corrode metals like brass over time. Studies have shown that ceramic nozzles offer the highest durability and chemical resistance, with lifespan ratings of 90 to 130 times longer than brass. Plastic nozzles, although

variable in performance, can be acceptable for cost-sensitive applications provided they are made of chemically resistant polymers. As a result, we recommend the use of ceramic nozzles for durability and chemical safety, while chemical-grade plastic nozzles may be used when budget constraints demand it [93, 94].

3.3.4 Conclusion and Final Specifications

Based on the geometric analysis and material considerations, the optimal nozzle configuration for the precision spraying system is as follows:

- **Nozzle Type:** Full cone nozzles should be used for both top and leg placements to ensure uniform saturation of the plant canopy.
- **Material:** Ceramic nozzles are the primary recommendation due to their high durability and resistance to chemical corrosion from fungicides like Curzate M. Chemical-grade plastic nozzles are a viable secondary option for budget-conscious implementations.
- **Spray Angle:** The calculated required angles are approximately 146.6°, 151.7°, and 147.9°. The best practical option is to select a single type of nozzle with a standard cone angle of **150°** for all placements. This simplifies procurement while providing coverage that closely matches all requirements.
- **Top Nozzle Inclination:** 0° (aimed directly downwards).
- **Leg-Mounted Nozzle Inclination:**
    - **Vertical:** Angled upwards at approximately **9.7°**.
    - **Horizontal:** Angled sideways at approximately **11.5°**.

This configuration ensures complete and uniform coverage of the target crop (see Figure 3.18), maximizing the effectiveness of the treatment while adhering to safety and durability standards.

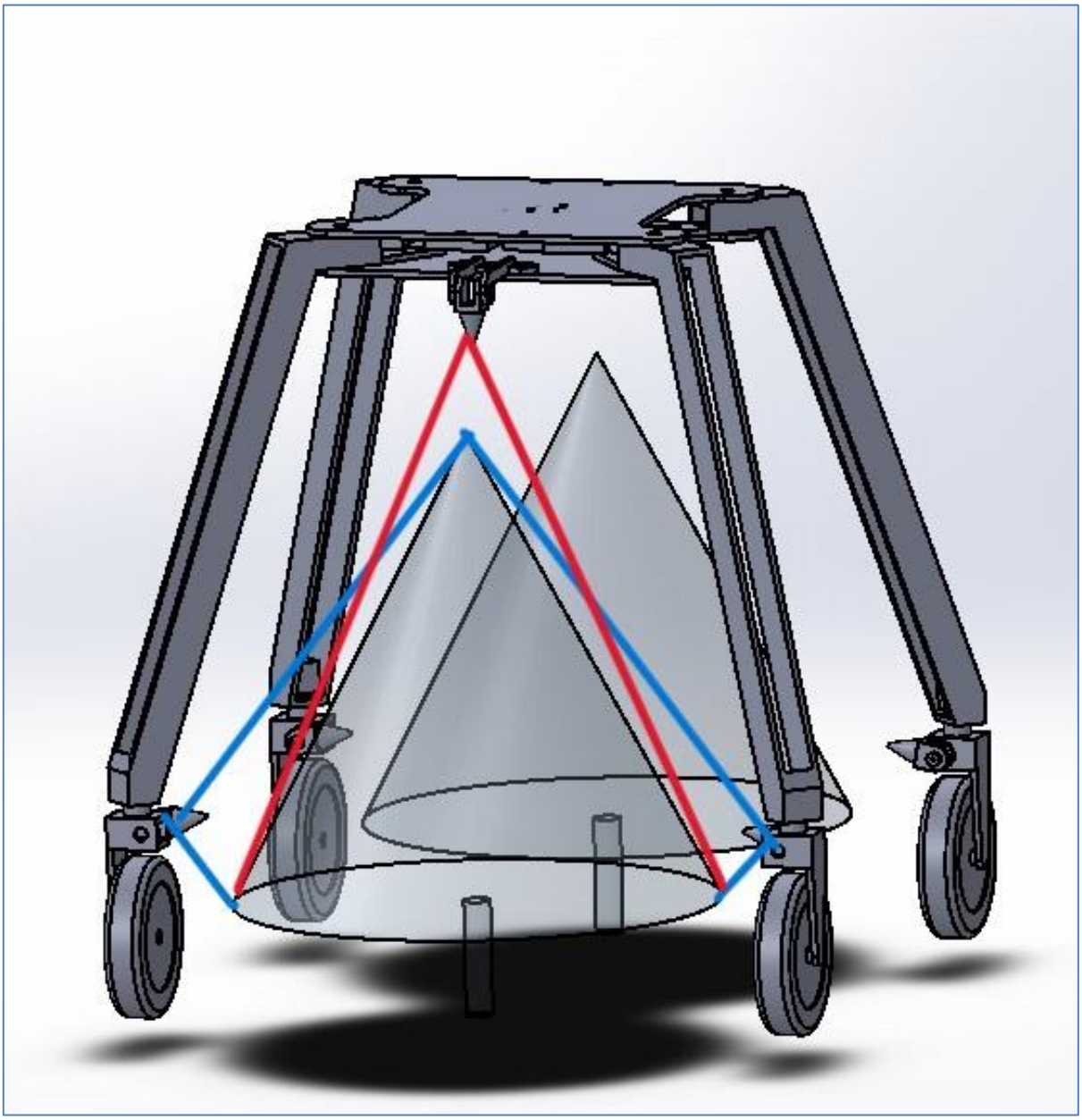


Figure 3.18: *The complete coverage of the target crop with the spraying system*

3.4 Web user Interface

The user interface serves as the primary point of interaction between the farmer and the underlying technologies. Developed using HTML, CSS, and JavaScript, the web-based interface is designed to be lightweight, accessible, and user-friendly across various devices. It enables the farmer to visualize real-time disease detection results and weather-based blight risk forecasts without needing to directly manipulate ROS commands or logic. By centralizing data display and control functions, the interface enhances decision-making and simplifies the management of plant health in the field.

3.4.1 Objectives

The main objective of the web interface is to provide the farmer with a clear, real-time view of the plant health and environmental conditions, while also enabling direct control over the system’s actions.

To achieve this, the interface is designed to display a range of critical information. This includes the direct output of the YOLO deep learning model, which reports whether infected areas are present, and a corresponding grid-like image gallery for visual confirmation of any detected infected leaves. It also provides a 24-hour weather-based blight risk forecast to notify the farmer of upcoming risks, alongside a real-time display of current weather data, including temperature, humidity, and wind.

In addition to its display capabilities, the interface provides comprehensive control over the system's functions. The user can start or stop the system, enable or disable the camera feed for the detection node, and select the desired spraying mode, choosing between "automatic" and "manual" operation. Furthermore, the interface allows for detailed mission control, giving the user the ability to accept or decline spraying suggestions, cancel a previously scheduled spray, or issue a direct, immediate command to begin spraying.

3.4.2 Architecture of the Interface

The web interface was developed to serve as the main communication layer between the farmer, the designed ROS system (AI-based detection system, weather forecast services, and robotic spraying unit). The primary communication mechanism enables the web interface to publish commands to ROS topics that our system subscribes to (typically command data), while simultaneously subscribing to topics where ROS publishes information (typically display data). This ensures a smooth bidirectional data flow between the system and its interface.

This real-time communication is made possible through WebSocket connections established via rosbridge, which creates a persistent, full-duplex communication channel between the browser and ROS. A WebSocket is an advanced communication protocol providing full-duplex communication channels over a single TCP connection, enabling real-time data exchange between web clients and servers. In our system, JavaScript code initiates a WebSocket connection to the rosbridge_server running on port 9090, acting as the bridge between ROS topics and the web interface.

3.4.3 Front-end Design and Layout

The frontend layout is structured into several functional sections, each dedicated to a specific task—for instance, the sidebar handles robot control settings. This modular design creates a clear, grid-based interface that enhances usability and ensures efficient access to all system features, resulting in the grid-like view for the web application shown in (Figure 3.19).

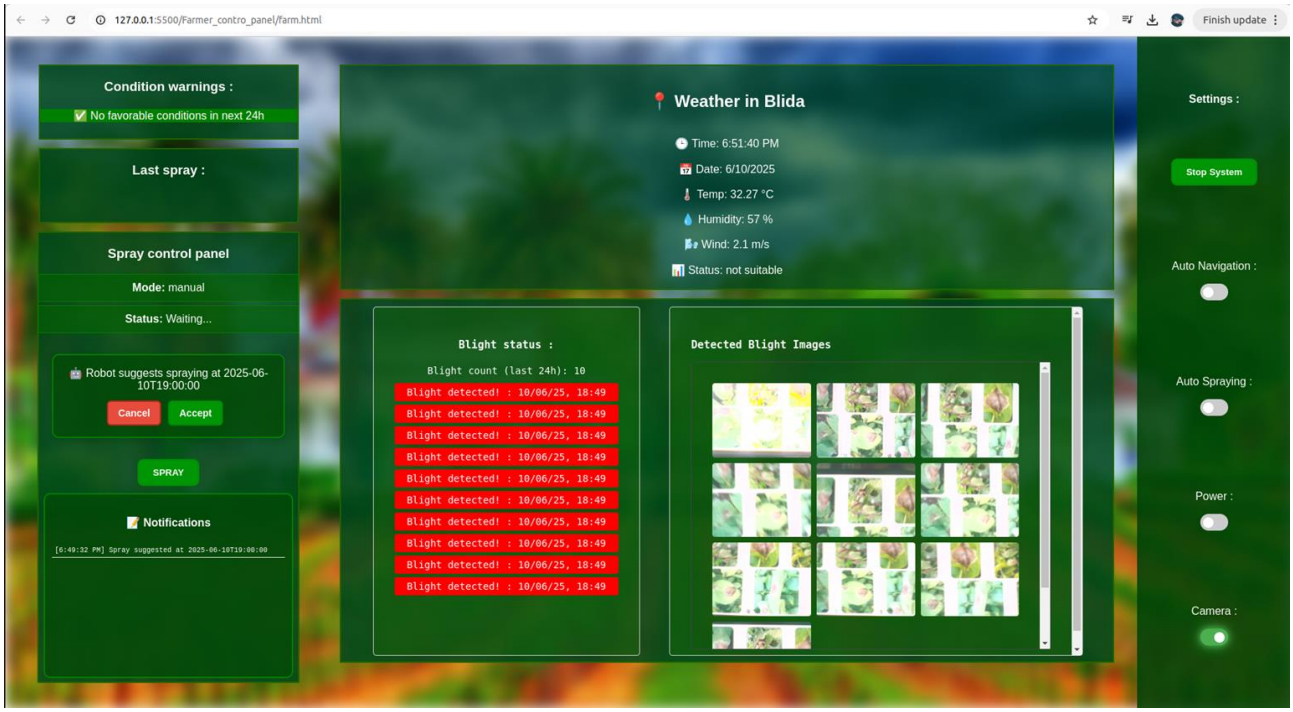


Figure 3.19: *The main dashboard of the web user interface, showing real-time data and control panels.*

a. *Sidebar Section*

The sidebar, shown in (Figure 3.20), contains all robot control parameters in the form of buttons and toggles. A green glow appears when a parameter is activated and disappears when it's turned off. These settings are sent to the robot to define its current permissions and operational behavior. The farmer has the ability to turn the robot's camera on or off (pausing the blight detector node), start or stop the system, set the spraying mode to manual or automatic, and control the auto-navigation and main power.

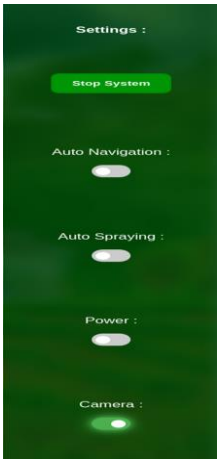


Figure 3.20: *The sidebar settings panel for robot control.*

b. *Weather Section*

This section, shown at the top of the dashboard in (Figure 3.21), streams real-time data for date, time, temperature, humidity, and wind. The data is specific to a selected city (Blida), determined by its longitude and latitude coordinates in the “weather_reporter” node. This provides the user with an immediate, at-a-glance view of the current environmental conditions.

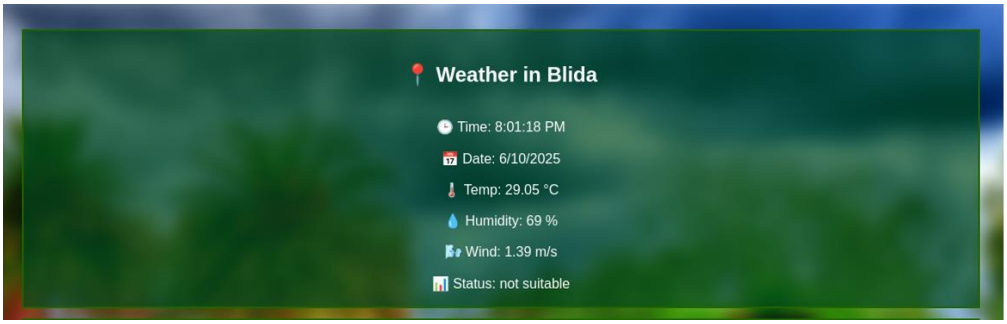


Figure 3.21: *The weather section displaying current conditions.*

c. *Blight Weather Conditions Warnings Section*

As the name suggests, this section shows the soonest meteorological danger so that the farmer can be aware of favorable blight conditions coming at a specific time within the next 24 hours. As shown in (Figure 3.22), if a risk is identified, a red warning box appears with the date and time of the predicted event.

Figure 3.22: *The condition warnings panel displaying a future blight risk.*

d. *Spray Control Section*

This section, shown in (Figure 3.23), displays all information related to spraying. This includes the current spraying mode (e.g., manual), the spraying status (e.g., waiting), and notifications for spraying commands, such as suggestions, cancellations, and acceptances. It also provides the user with the ability to cancel or accept any suggested or scheduled spraying action, as well as launch an unconditional spray action from a separate "SPRAY" button.

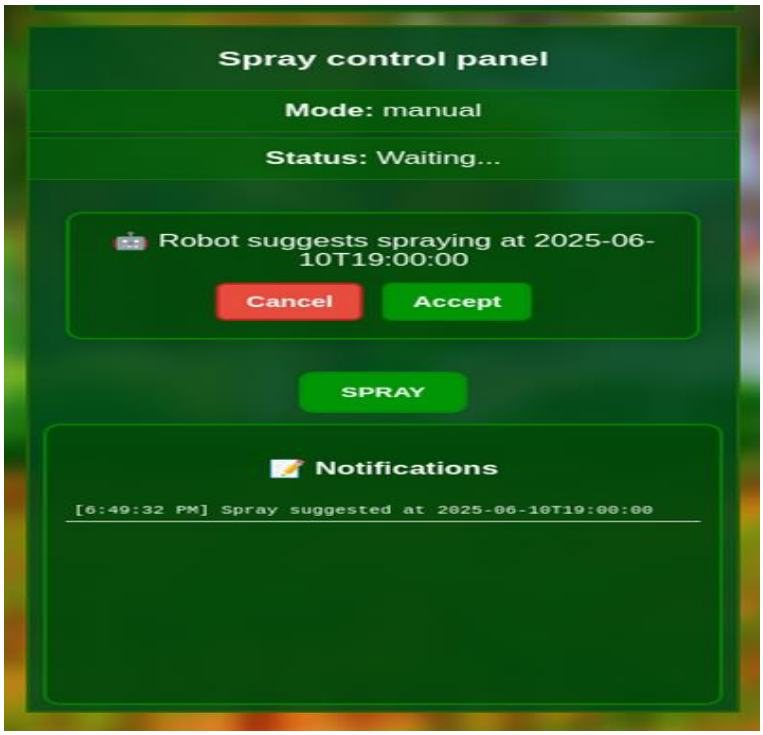


Figure 3.23: *The spray control panel displaying a spraying suggestion.*

e. *Spray Logs Section*

This section displays recent spraying activity. As shown in (Figure 3.24), it displays the exact time and date of the last spray committed, providing a clear history for the farmer to reference.

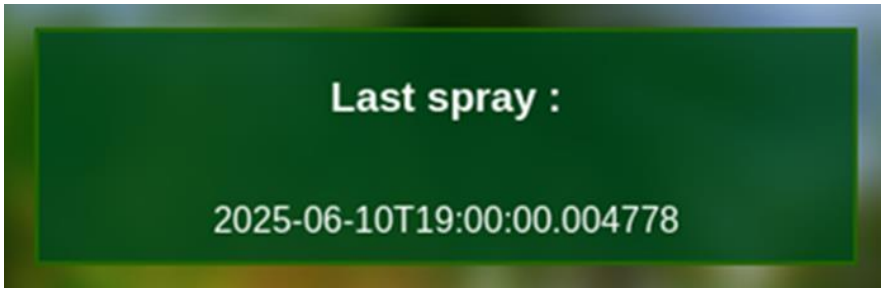


Figure 3.24: *The spray log panel showing the timestamp of the last completed spray.*

f. *Blight Section*

This section (Figure 3.25) displays data from the YOLO prediction node in ROS, showing alerts when blight is detected along with the detection time. It includes a counter to track the number of detections, helping assess disease spread, and a gallery of detected blight images on leaves for visual verification.

Figure 3.25: *The blight status panel showing detection logs and an image gallery.*

3.4.4 Backend Communication and Integration

The web interface is not a standalone application but a tightly integrated component of the overall system, bridging the gap between the farmer and the intelligent robotic platform. It communicates with the backend components—particularly the ROS 2 environment through asynchronous and websocket interactions. These connections allow the interface to both retrieve information and send user commands in real time.

a. *Real-Time WebSocket Bridge with ROS 2*

To enable real-time, low-latency communication between the ROS 2 environment and the browser, a WebSocket bridge is implemented using rosbridge_server, which offers live streaming of the data published by our system's nodes. In our setup, as shown in (Figure 3.26), a ros_connection.js JavaScript file instantly requests to connect to the WebSocket channel as soon as the page loads. The rosbridge_server is launched in the ROS environment and starts a WebSocket server on port 9090 (Figure 3.27), acting as the bridge between ROS topics and the web interface.

Figure 3.26: *JavaScript connection to WebSocket.*

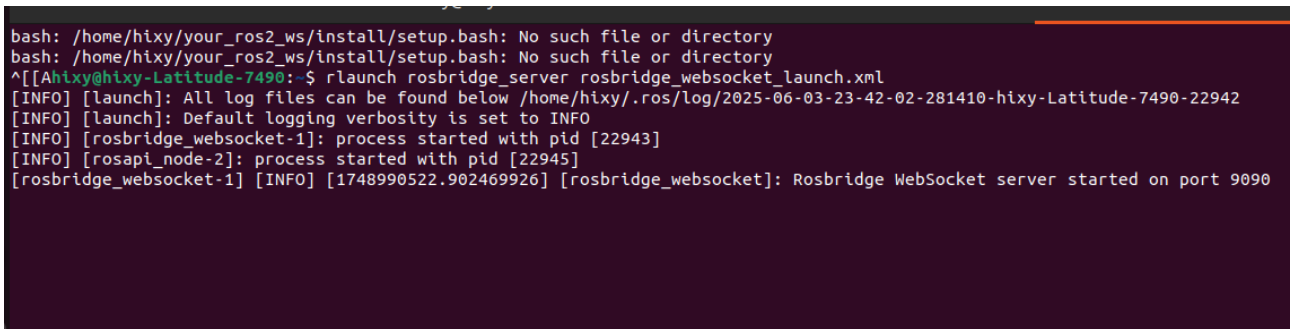


Figure 3.27: *Launching the rosbridge server.*

b. *Blight Display*

For the blight display, a dedicated blight.js JavaScript file is created. As shown in the data flow diagram in (Figure 3.28), this script holds the responsibility of updating the front-end with the latest updates from the blight_detector node. It does this by subscribing to the /blight_detection topic for textual alerts and the /blight_image topic for the visual data used to populate the image gallery.

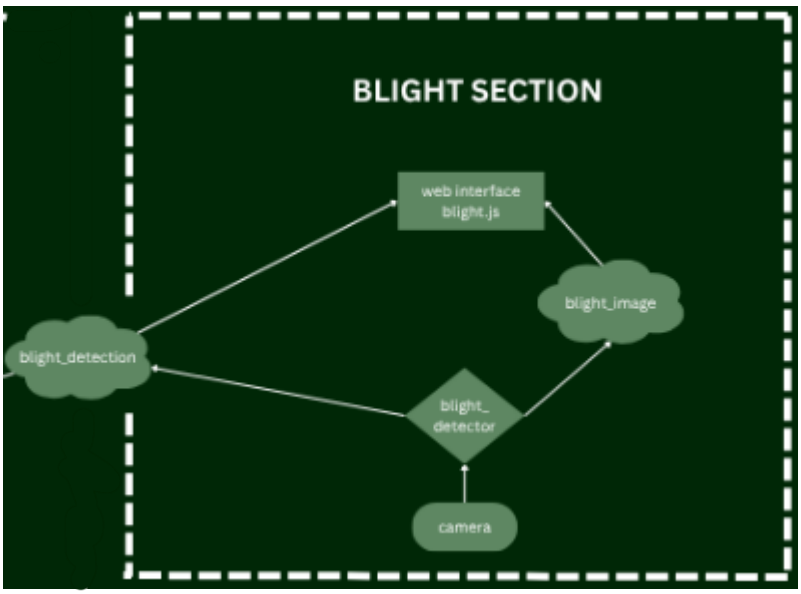


Figure 3.28: *Data flow for the blight display section.*

c. *Weather Data Retrieval and Integration*

As shown in (Figure 3.29), the interface retrieves real-time and forecast weather data through a weather.js JavaScript file. This script subscribes to the /weather_status topic and listens for data published by the weather_reporter node. The front-end then uses this data to populate both the main weather section and the weather warnings section, providing a 24-hour forecast of potential blight danger.

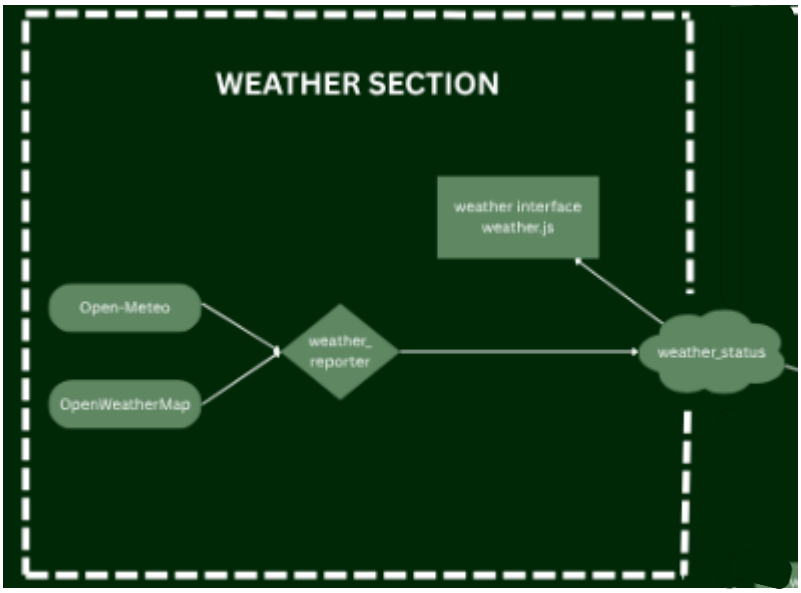


Figure 3.29: *Data flow for the weather section.*

d. *Spray Log Display*

For the spray log, a spray.js JavaScript file subscribes to the /spray_log topic. It listens to the data published on this topic to find and display the timestamp of the latest completed spray action, providing a clear history of interventions.

e. *Spray Control*

The spray control functionality is a central feature that involves multiple JavaScript files and ROS topics.

- **spray_modes.js**: This file handles user selections for the spraying mode ("manual" or "auto") and publishes the choice to the /spraying_mode topic. It also publishes the user's response to spray suggestions (accept/decline) to the /spray_accept and /spray_cancel topics.
- **spray.js**: This file manages the unconditional spray command. When the user clicks the "SPRAY" button, it publishes a message to the /spray_command topic.

- **spray_notifications.js**: This script is responsible for displaying system actions, such as schedules and suggestions, as well as user responses. It subscribes to the /spray_schedule_notification topic to receive the necessary data for display.

All these JavaScript modules are connected and managed by the main farm.html file, creating a cohesive and interactive user experience. The complete architecture, illustrating the connections between all ROS nodes, web interface files, and topics, is shown in (Figure 3.30).

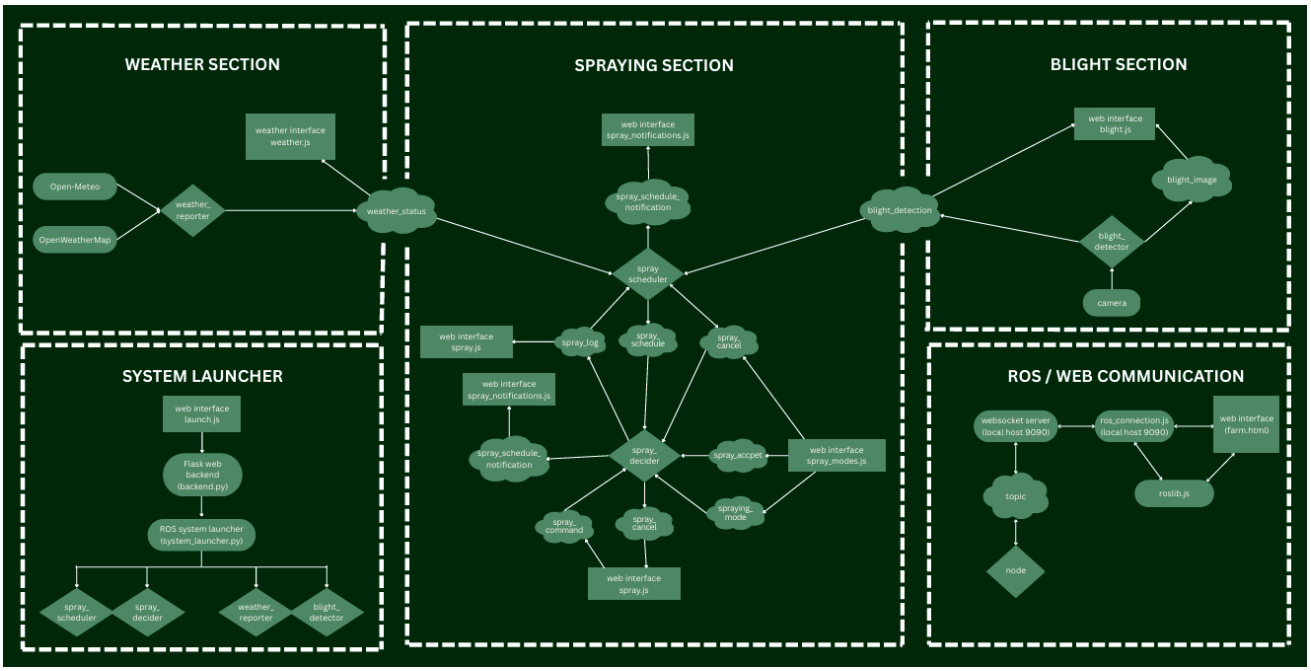


Figure 3.30: *A comprehensive block diagram of the full system architecture, detailing the interaction between all ROS nodes and the web interface components.*

3.4.5 System Launch Setup

The system is a website that connects to a ROS 2 system consisting of four nodes via a rosbridge. To run this system, there are two primary methods: a manual setup and a more user-friendly, automated setup using a Flask backend.

a. *Manual Setup*

The manual method requires launching each component from the terminal.

1. **Initiate the WebSocket Bridge**: The connection between ROS and the web browser is established by running the rosbridge_server with the following command: rlaunch rosbridge_server rosbridge_websocket_launch.xml
2. **Open the Web Interface**: Open the farm.html file in a web browser. This will automatically attempt to connect to the ROS system via the WebSocket.
3. **Launch ROS Nodes**: The four main ROS nodes can be launched all at once using a ROS launcher or individually. To run a single node, use the command: ros2 run <pkg_name> <node_name> (e.g., ros2 run weather_reporter weather_node)

If ROS 2 is installed and configured correctly, the system will run effectively with this method.

b. *Automated Setup with Flask Backend*

To eliminate the need for manual terminal commands and make the system more user-friendly, a Flask backend web server was integrated. This allows the user to start the entire system simply by opening the website and pressing the "Start System" button.
The Flask application is designed to launch the main ROS launcher file but waits for a command from the web interface. This interaction works as follows:

- The Flask application listens for HTTP POST requests on two specific endpoints: http://localhost:5000/start and http://localhost:5000/stop.
- A backend.js file in the web interface sends a POST request to the appropriate endpoint when the user clicks the "Start System" or "Stop System" button.

- When a request is received on the /start channel, the Flask server executes the ros_launcher, which starts all four ROS nodes simultaneously.
- When a request is received on the /stop channel, the Flask server uses the psutil library to find the Process ID (PID) of the main ROS launcher process and all of its children (the running nodes) and then terminates them all.

To make the system completely seamless, the final step is to configure the rosbridge WebSocket and the Flask backend to run instantly when the operating system boots. This allows the system to run silently in the background until the user opens the website and starts it from the interface.

c. *Conclusion*

The interface enhances the system's usability by presenting real-time data and controls in a clear, accessible layout. It allows the user to monitor blight detection, weather forecasts, and spraying history, while also enabling direct control of the system. By centralizing information and actions in one place, the interface empowers the user to make timely, informed decisions and manage crop health more effectively.

Chapter 4 Validation and Results

4.1 System Performance Analysis

4.1.1 The System Usability with the User Interface

The overall usability of the system was evaluated by testing the web interface, which serves as the primary control and monitoring platform for all underlying components. Through extensive testing and observation, the following key features were verified, confirming the system's intuitive and effective design:

a. *Weather Monitoring and Auto-Update Functionality*

- The system successfully fetches real-time weather data at 10-minute intervals, which is displayed in the main weather panel (Figure 4.1a).
- Favorable conditions for late blight development are accurately detected and displayed in real time.
- The interface clearly distinguishes between current and forecast weather risks using color-coded warnings and status indicators. For instance, a green panel indicates no immediate threat (Figure 4.1b), while a red warning highlights an upcoming period of high risk (Figure 4.1c).

(a)

(b)

(c)

Figure 4.1: *The weather monitoring and warning panels of the user interface, with (a) Real-Time Weather Panel, (b) "Next Suitable Conditions" Warning, (c) "No Favorable Conditions" Warning.*

b. *Blight Detection Integration*

- The blight detection node is triggered through a camera control toggle on the interface.
- Captured images of infected crops are correctly analyzed and uploaded to the gallery section with proper labeling.
- Detections are timestamped and displayed, allowing users to track and review incidents.
- A separate window provides a live camera feed with real-time bounding boxes drawn around detected blight and leaves, as shown in (Figure 4.2).

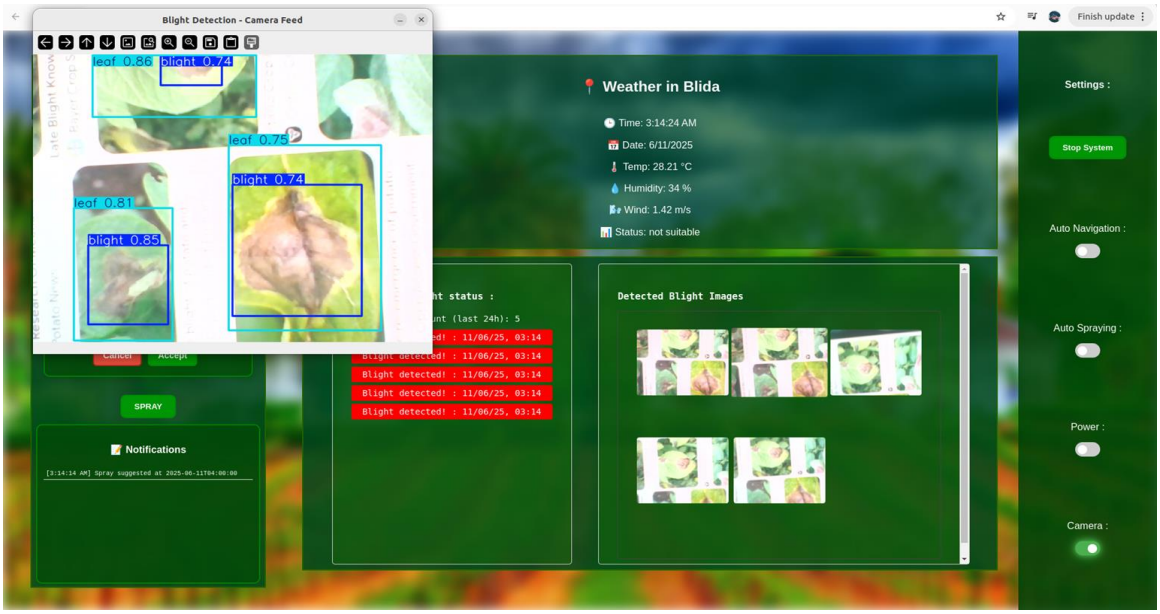
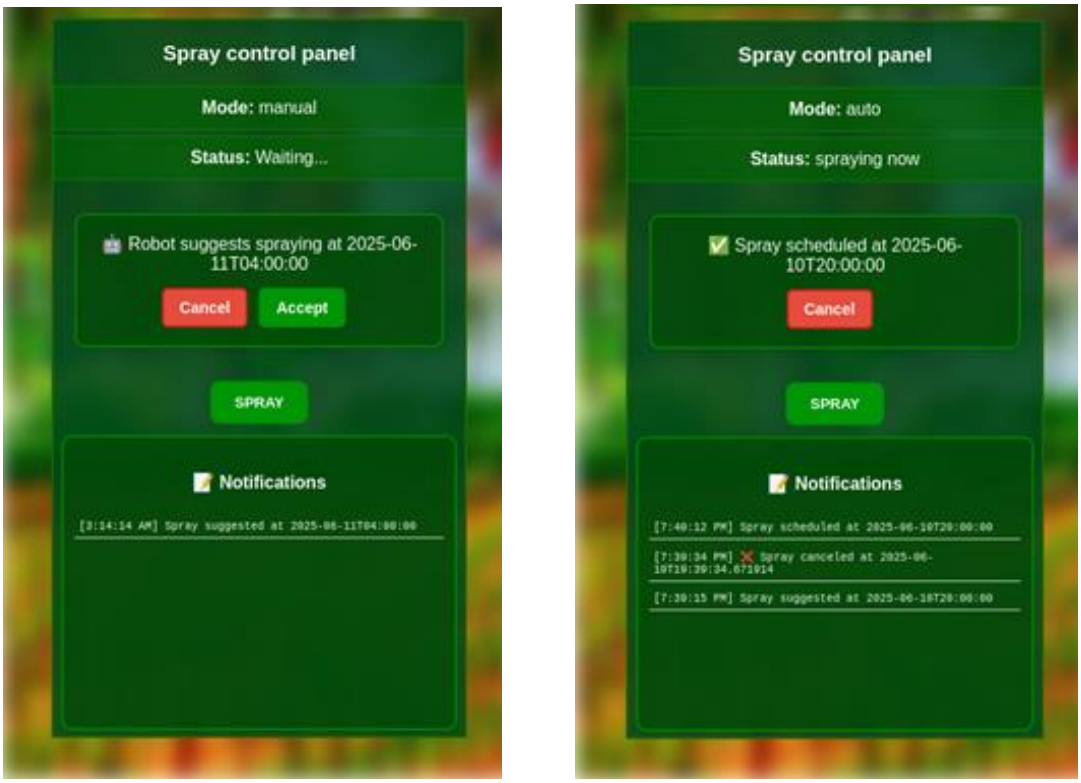


Figure 4.2: *Real-time blight detection feed showing bounding boxes around infected areas.*

c. *Spray Control Modes*

- **Automatic Spraying Mode:** The system autonomously schedules a spray event when either favorable conditions for late blight are detected or an infection is confirmed by the AI detector. The interface then shows the scheduled spray, as seen in (Figure 4.3b).
- **Manual Spraying Mode:** The user is prompted to confirm or decline any spray recommendation (Figure 4.3a). If accepted, the system schedules the spray accordingly.



(a) ***(b)***

Figure 4.3: *Spray control panel showing (a) a user prompt in Manual Mode and (b) an automatically scheduled spray in Auto Mode.*

d. *Spraying Status and Scheduling*

- Scheduled spraying actions are tracked and displayed in the interface. Once a user accepts a manual suggestion, the interface updates to reflect the officially scheduled mission (Figure 4.4).
- When a spray event reaches its execution time, the system logs the operation and updates the spray history table.
- Users are informed of past spraying activities within a 15-day window for reference.

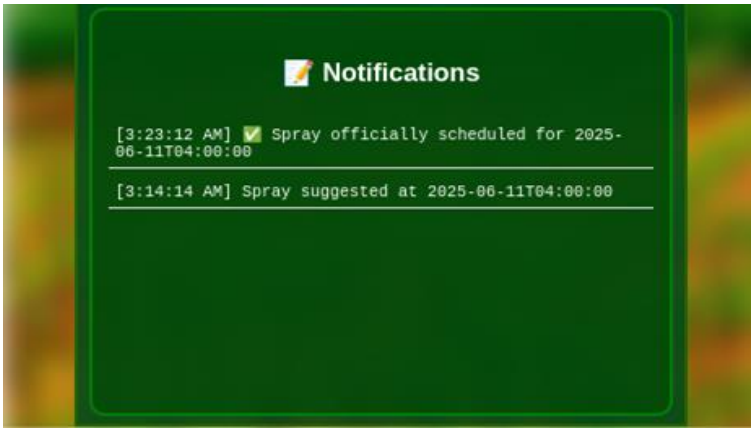


Figure 4.4: *The spray control panel after a user accepts a suggestion in manual mode, showing the mission is now officially scheduled.*

e. *System and Node Control*

- **Camera Control:** The camera toggle button activates or deactivates the blight detection node in real time.
- **System Control:** The "Start System" and "Stop System" buttons control all nodes. Clicking "Stop System" safely shuts down all active services, while "Start System" re-initializes them. The control panel for these actions is shown in Figure 4.5.

Figure 4.5: *The sidebar control panel for system and node settings.*

f. *General Observations*

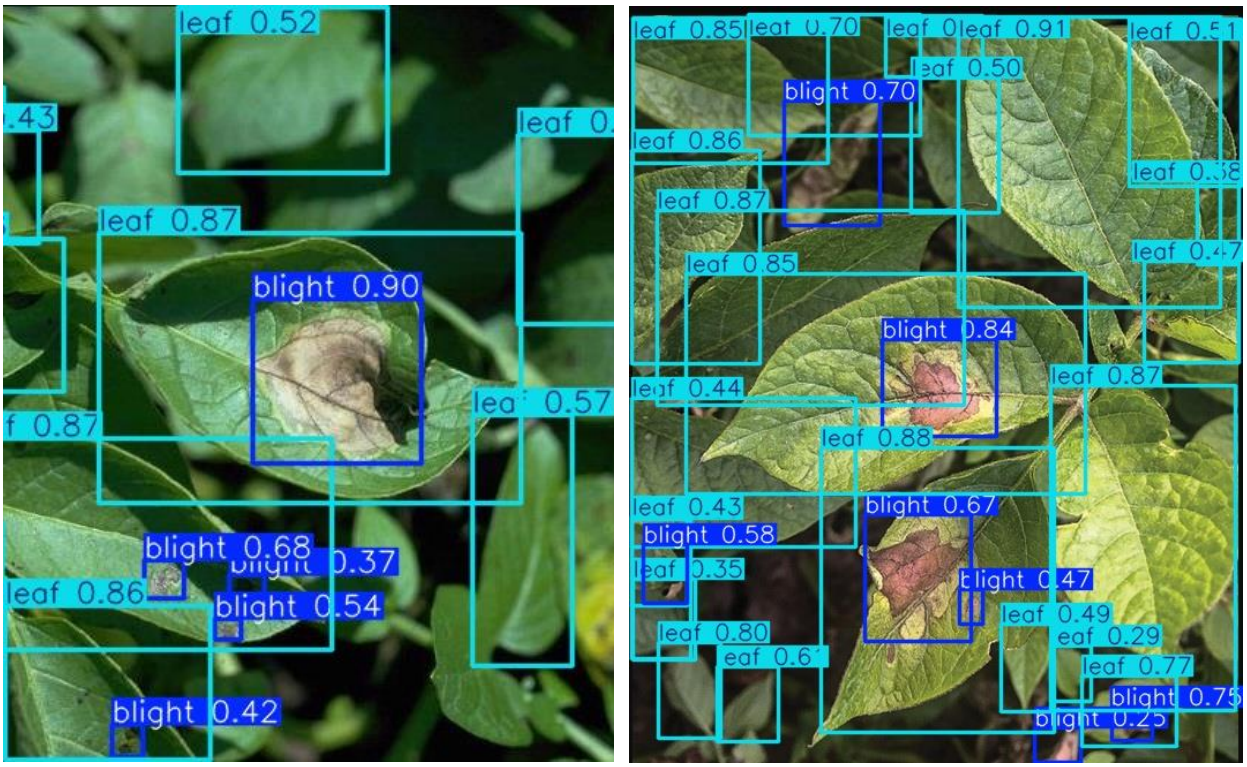
- All functional objectives of the web interface have been met.
- The user experience is responsive, with smooth transitions between system states.
- The interface provides centralized and accessible control over weather analysis, disease monitoring, and spray decisions.

4.1.2 YOLO Detection Performance Evaluation

a. *Generalization on Unseen Internet Images*

After training, the model's generalization capabilities were evaluated using a set of unseen images collected from the internet, which were not part of the training dataset. The model demonstrated high effectiveness, accurately detecting nearly all instances of blight present in these images.

A key strength observed was its ability to identify challenging cases, including both small and blurred infections (Figure 4.6). Furthermore, the model proved robust against false positives by accurately avoiding the detection of soil (Figure 4.7). Critically, it successfully differentiated late blight from visually similar diseases, such as Botrytis gray mold (Figure 4.8). Overall, these evaluations confirmed the model's high detection accuracy on diverse, real-world images, even in challenging backgrounds.



(a) (b)

Figure 4.6: *Detection of (a) small and (b) blured blight infections*

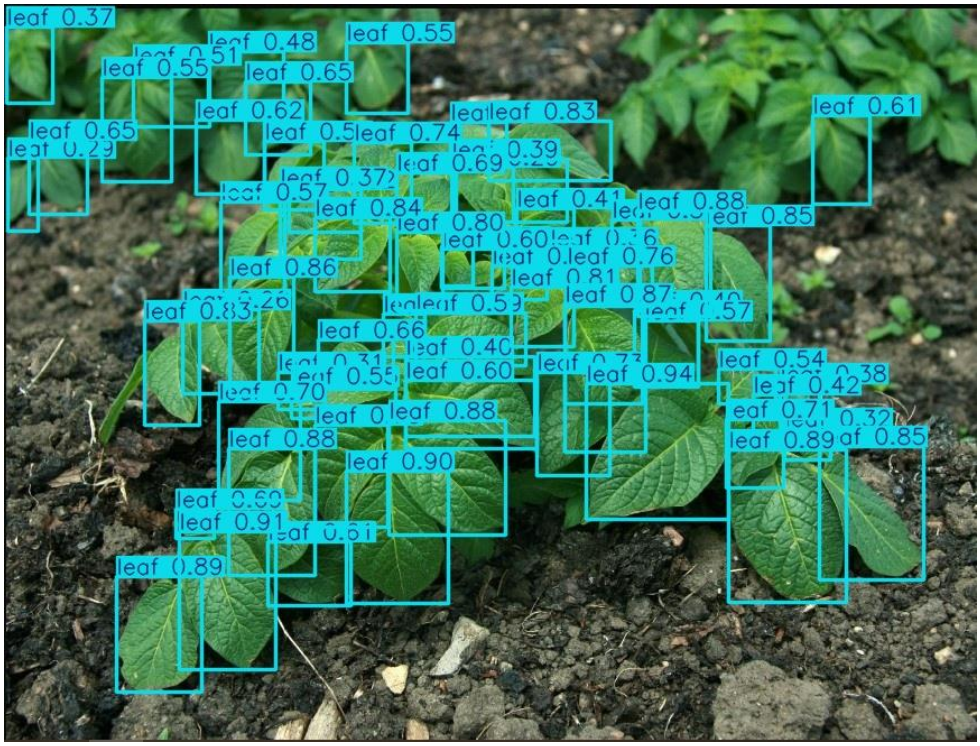


Figure 4.7: *Detection of healthy leaves in the presence of soil*

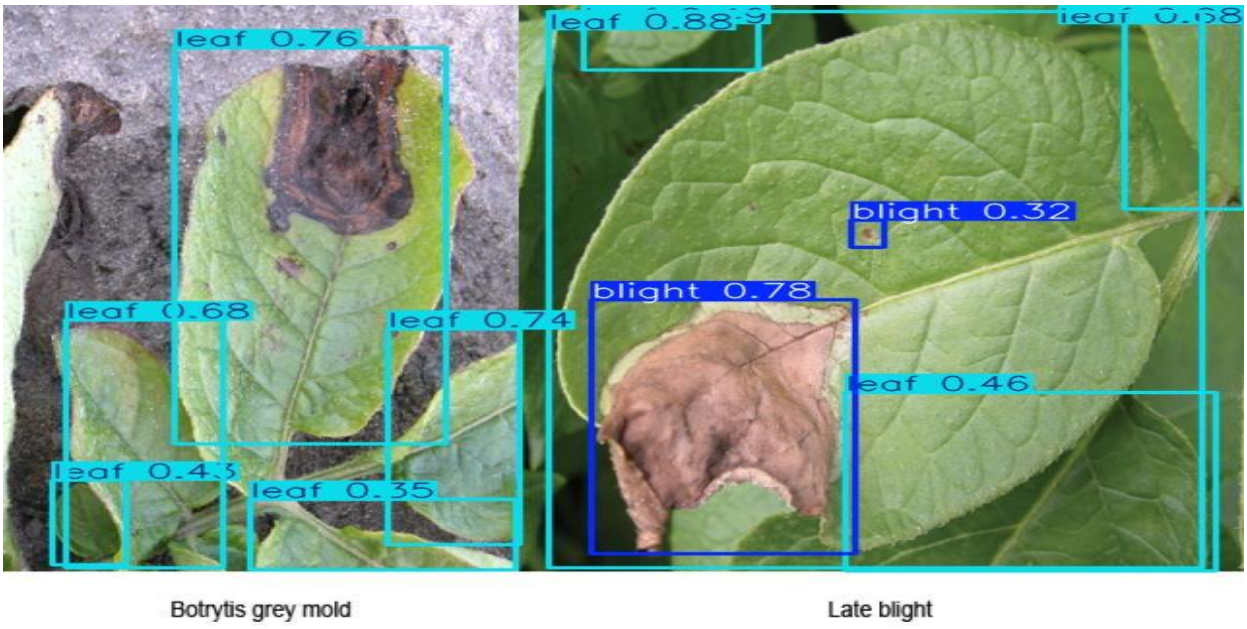


Figure 4.8: *Differentiation of Late Blight from Botrytis gray mold*

b. *Real-Time Detection on images in Mobile phone*

After the deployment of the model on ROS, real-time detection was tested using a camera feed with images from a mobile phone. This testing evaluated the rapidity and accuracy of the model, particularly with low-quality images, yielding very good results. An example of these real-time detections is shown in (Figure 4.9). Despite this strong performance, certain limitations were observed. For instance, in some cases, the model had a tendency to misclassify early blight as late blight, as seen in (Figure 4.10: Misclassification of Early Blight as Late Blight). This occurred due to the significant visual similarity between their symptoms. The accuracy remained good, and blight was still correctly detected even with poor imaging resolution.

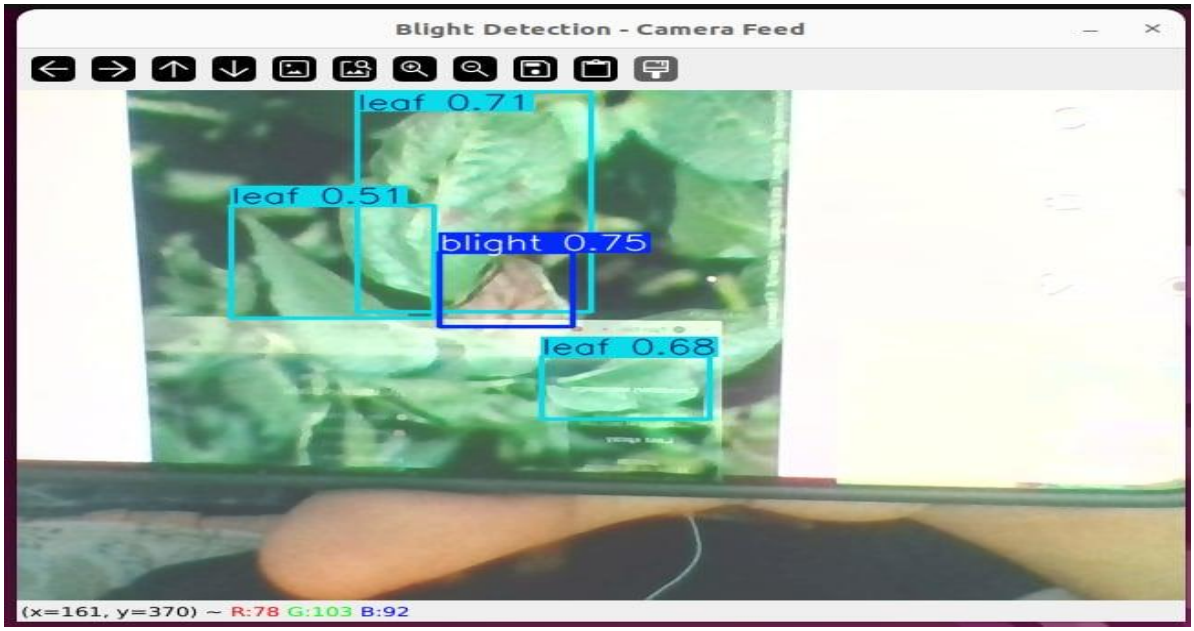


Figure 4.9: *Real-time Detections by the YOLOv8n Model Showing Late Blight and General Good Detection*

Figure 4.10: **Misclassification of Early Blight as Late Blight**

c. *Field Deployment and On-Site Evaluation*

For further real-world validation, the system was put to the test in an agricultural environment by visiting **ITCMI** (Institut Technique des Cultures Maraîchères Et Industrielles). However, as all potato crops had been removed, the system was tested on a morphologically similar plant species. During this field test, the model successfully detected healthy leaves (Figure 4.11). Furthermore, a disease very similar to late blight was encountered on site, and the model successfully detected it as well (Figure 4.12).

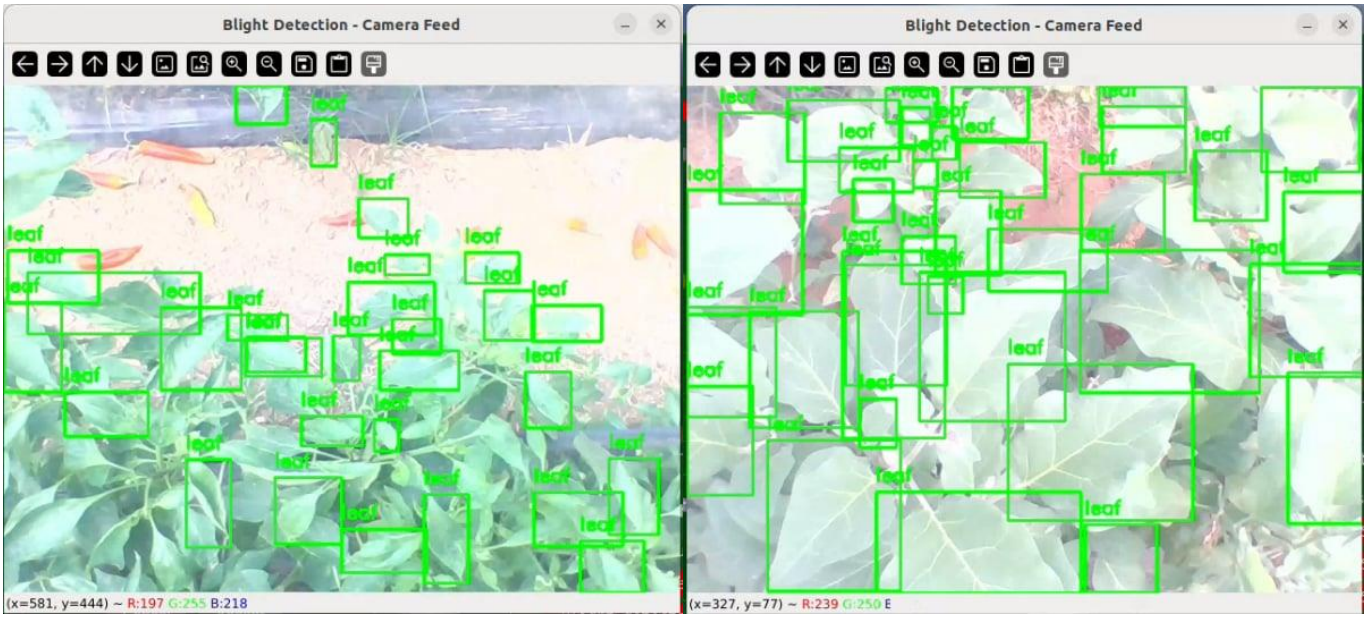


Figure 4.11: Detection of healthy leaves in field conditions

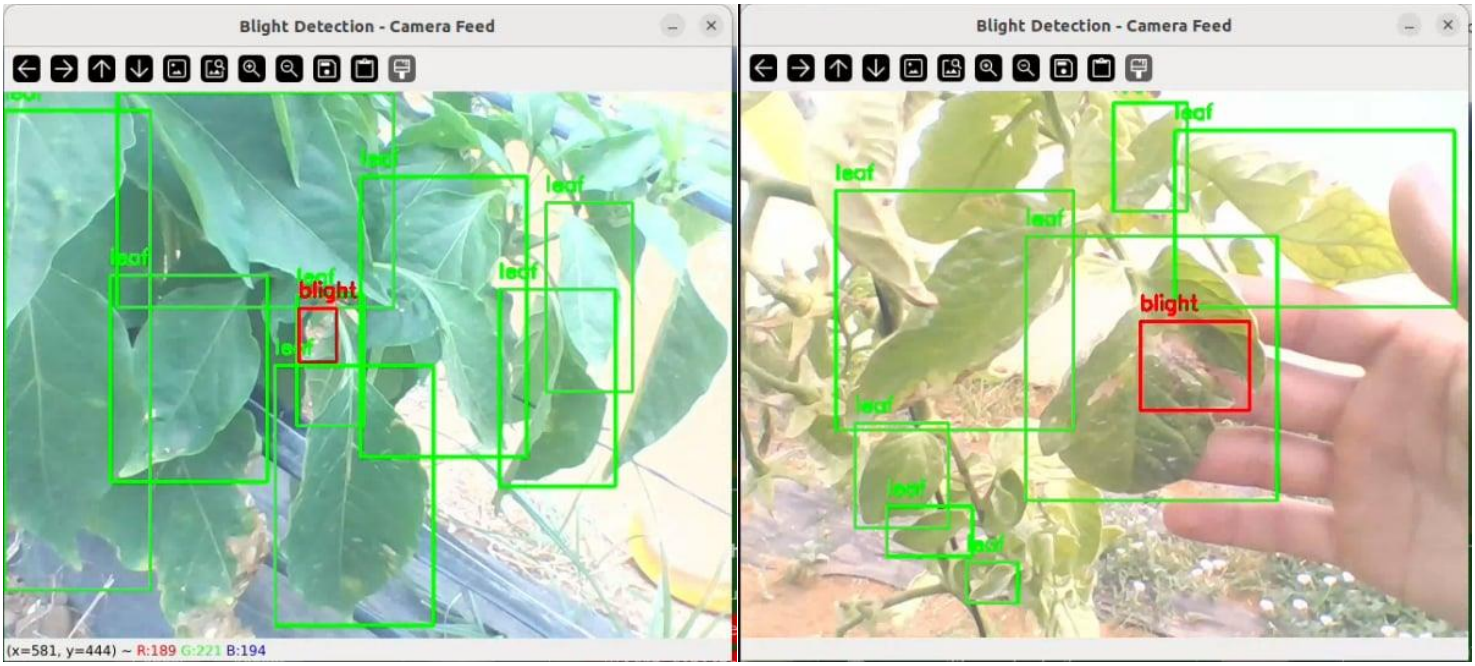


Figure 4.12: *Detection of a late blight-like disease in the field*

4.2 Limitations and Observations

This section outlines the current limitations identified during the development and testing of the intelligent decision-support system, along with general observations. It also proposes avenues for future improvements to enhance the system's robustness, efficiency, and practical applicability.

4.2.1 Environmental Constraints

- **Variability in Field Conditions**: The performance of the AI detection model, while robust, can still be influenced by extreme variations in real-world agricultural environments, such as inconsistent lighting (e.g., direct sunlight, shadows), diverse leaf orientations, and the presence of dust or debris that might mimic or obscure disease symptoms.
- **Microclimatic Specificity**: While the weather-based prediction system utilizes high-resolution data, localized microclimates within a large farm might exhibit slight deviations from broader regional forecasts, potentially affecting the precise timing of blight predictions for very small areas.

4.2.2 System Bottlenecks

- **Computational Intensity for AI Training**: The initial training of the deep learning model, particularly with large and diverse datasets, remains computationally intensive, requiring significant GPU resources. While inference on edge devices is optimized, the training phase can be a bottleneck for rapid model iteration.
- **Dataset Generalization**: Despite efforts to create a hybrid dataset, the model's ability to generalize perfectly to every conceivable real-world scenario (e.g., highly unusual blight manifestations, different potato varieties) is an ongoing challenge that requires continuous data acquisition and model retraining.
- **Physical Robot Mobility**: The current robotic platform, while adaptable, might face limitations in extremely rough terrain or highly dense crop fields where physical obstructions could impede seamless navigation or precise spraying.

4.3 Future Improvements

4.3.1 Advanced AI Model Refinement:

- **Few-shot Learning**: Investigate few-shot learning techniques to enable the model to detect new or rare plant diseases with minimal training data, improving adaptability.
- **Enhanced Small Object Detection**: Further optimize the YOLOv8n model or explore other architectures to improve the detection of very small, incipient lesions that are crucial for even earlier intervention.
- **Multi-disease Detection**: Expand the model's capabilities to detect and differentiate multiple potato diseases beyond late blight, offering a more comprehensive plant health monitoring solution.

4.3.2 Sophisticated Weather Prediction:

- **Hyperlocal Weather Stations**: Integrate the system with on-site, hyperlocal weather stations within the farm to gather more precise microclimatic data, leading to even more accurate and localized blight forecasts.
- **Predictive Analytics for Treatment Efficacy**: Develop algorithms that not only predict disease outbreaks but also recommend specific fungicide types or application rates based on predicted disease severity and environmental conditions.

4.3.3 Robotic Platform Enhancements:

- **Autonomous Navigation**: Implement more advanced autonomous navigation capabilities, potentially using Deep Reinforcement Learning, to allow the robot to learn optimal paths, avoid obstacles, and navigate complex field geometries more efficiently.
- **Dynamic Spraying Mechanism**: Explore a dynamic spraying mechanism that can adjust spray angles and pressure in real-time based on plant density, height, and precise lesion location, further optimizing chemical application.
- **Energy Efficiency**: Research and integrate more energy-efficient components and power management strategies to extend the robot's operational time in the field.

4.3.4 User Interface Evolution:

- **Advanced Reporting**: Develop more comprehensive reporting features, including historical trends of disease incidence, treatment effectiveness, and detailed environmental data analysis.
- **Mobile Application**: Create a dedicated mobile application for the system, providing farmers with on-the-go access to critical information and control functionalities.
  - **Scalability and Multi-robot Systems**:
- **Fleet Management**: Investigate the potential for managing a fleet of multiple robots working collaboratively across larger agricultural areas, optimizing coverage and efficiency.
- **Data Integration with Farm Management Systems**: Explore integration with existing farm management systems to provide a holistic view of agricultural operations.

General Conclusion

In this thesis, we addressed the problem of optimizing plant disease management and reducing agrochemical usage, particularly for late blight in potato crops, which remain a significant challenge for farmers in Algeria and globally. Our goal was to develop an intelligent decision-support system that combines early disease detection with predictive weather forecasting to enable precise and timely interventions, thereby enhancing agricultural sustainability.

Through the implementation of a deep learning-based AI model (YOLOv8n) for real-time disease detection, integration of a weather-based prediction system using meteorological APIs, and the design of a precision spraying robotic platform managed via an intuitive web interface and successfully demonstrated an integrated system capable of proactive threat mitigation and automated precision spraying. The developed system empowers farmers with data-driven insights and remote control over crop health management, leading to more informed decisions and reduced unnecessary chemical applications. Our findings demonstrate that a hybrid dataset strategy, combined with optimized lightweight deep learning models like YOLOv8n and systematic parameter tuning, is crucial for achieving effective and adaptable performance in real-world agricultural settings.

This research contributes to the field of smart agriculture and agricultural robotics by proposing a novel integrated decision-support system specifically tailored for late blight management in potato crops. It improves the precision of disease detection through a robust AI model capable of operating on edge devices, and enhances intervention timing by integrating real-time weather forecasting with a recommendation engine. While the system shows promising results, certain limitations such as the need for extensive computational resources during initial model training and the ongoing challenge of

dataset generalization to all real-world agricultural variabilities may be addressed in future studies.

The proposed approach can be applied in various agricultural settings for targeted plant disease management, paving the way for significant reductions in agrochemical usage, increased crop yields, enhanced food security, and a more sustainable farming future. For future work, the AI model can be further improved to detect symptoms at a very early stage and adapt to a wider range of environmental conditions, exploring advanced robotic navigation and control strategies using DRL, and conducting long-term field trials to validate the system's economic and environmental impact at scale.

Overall, this thesis lays the foundation for a new generation of intelligent and sustainable agricultural practices, particularly in regions like Algeria where advanced technological adoption in agriculture is still emerging

Annexes

Bibliography

[1] *Annual Report of the Bank of Algeria 2023*.

[2] Zhang N, Wang M, Wang N. Precision agriculture—a worldwide overview. *Comput Electron Agric* 2002; 36: 113–132.

[3] McBratney A, Whelan B, Ancev T, et al. Future Directions of Precision Agriculture. *Precis Agric* 2005; 6: 7–23.

[4] Pieterse L. Potato disease-detection technologies. *CHIPS* 2024; 38: 58–60.

[5] Tiwari S, Srivastava A. Farmers' knowledge and perception of late blight of potato and its management strategies in Kailali and Banke districts of Nepal. *Plant Pathol Quar* 2024; 14: 1–9.

[6] Sahu HP, Satapathy RR. Indigenous Technical Knowledge in Plant Disease Management. *Asian J Agric Ext Econ Sociol* 2021; 377–380.

[7] IITKA_Book_Traditional-Knowledge-in-Agriculture-English_0_0.pdf, https://icar.org.in/sites/default/files/2022-06/IITKA_Book_Traditional-Knowledge-in-Agriculture-English_0_0.pdf (accessed 4 June 2025).

[8] Nguyen H. *ARTIFICIAL INTELLIGENCE AND ITS IMPACT ON WORKFORCE. Job Clarification*. Undergraduate Thesis, Centria University of Applied Sciences, 2019.

[9] Rani R, Sahoo J, Bellamkonda S, et al. Role of Artificial Intelligence in Agriculture: An Analysis and Advancements With Focus on Plant Diseases. *IEEE Access* 2023; 11: 137999–138019.

[10] Aydın N, Erdem OA, Tekerek A. Comparative Analysis of Traditional Machine Learning and Transformer-based Deep Learning Models for Text Classification. *J Polytech* 2024; 1–1.

[11] *Pattern Classification and Scene Analysis* . Richard O. Duda , Peter E. Hart. *Libr Q* 1974; 44: 258–259.

[12] LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S. *Philos Trans R Soc Lond* 1763; 53: 370–418.

[13] Breiman L. Random Forests. *Mach Learn* 2001; 45: 5–32.

[14] Quinlan JR. Induction of decision trees. *Mach Learn* 1986; 1: 81–106.

[15] Cortes C, Vapnik V. Support-vector networks. *Mach Learn* 1995; 20: 273–297.

[16] Robbins H, Monro S. A Stochastic Approximation Method. *Ann Math Stat* 1951; 22: 400–407.

[17] Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition. *Proc IEEE* 1998; 86: 2278–2324.

[18] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, pp. 770–778.

[19] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, pp. 779–788.

[20] Pacal I, Kunduracioglu I, Alma MH, et al. A systematic review of deep learning techniques for plant diseases. *Artif Intell Rev* 2024; 57: 304.

[21] Tan M, Le QV. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Epub ahead of print 2019. DOI: 10.48550/ARXIV.1905.11946.

[22] Nayeri ZM, Ghafarian T, Javadi B. Application placement in Fog computing with AI approach: Taxonomy and a state of the art survey. *J Netw Comput Appl* 2021; 185: 103078.

[23] Cebollada S, Payá L, Flores M, et al. A state-of-the-art review on mobile robotics tasks using artificial intelligence and visual data. *Expert Syst Appl* 2021; 167: 114195.

[24] Hafiz AM. Image Classification by Reinforcement Learning with Two-State Q-Learning. Epub ahead of print 31 October 2020. DOI: 10.48550/arXiv.2007.01298.

[25] Sujatha R, Chatterjee JM, Jhanjhi N, et al. Performance of deep learning vs machine learning in plant leaf disease detection. *Microprocess Microsyst* 2021; 80: 103615.

[26] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. Epub ahead of print 2014. DOI: 10.48550/ARXIV.1409.1556.

[27] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the Inception Architecture for Computer Vision. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, pp. 2818–2826.

[28] Hasan RI, Yusuf SM, Alzubaidi L. Review of the State of the Art of Deep Learning for Plant Diseases: A Broad Analysis and Discussion. *Plants* 2020; 9: 1302.

[29] Natij Y, Maafiri A, El Karch H, et al. Plant Disease Recognition: A Comprehensive Mini Review. In: *2024 11th International Conference on Wireless Networks and Mobile Communications (WINCOM)*. Leeds, United Kingdom: IEEE, pp. 1–8.

[30] Shoaib M, Shah B, EI-Sappagh S, et al. An advanced deep learning models-based plant disease detection: A review of recent research. *Front Plant Sci* 2023; 14: 1158933.

[31] Luaibi AR, Salman TM, Miry AH. Detection of citrus leaf diseases using a deep learning technique. *Int J Electr Comput Eng IJECE* 2021; 11: 1719.

[32] Jafar A, Bibi N, Naqvi RA, et al. Revolutionizing agriculture with artificial intelligence: plant disease detection methods, applications, and their limitations. *Front Plant Sci* 2024; 15: 1356260.

[33] Husain SO, Jagadish S, Armoogum V, et al. Analysis of Machine Learning and Deep Learning Algorithms for Plant Disease Detection and Classification. In: *2024 Second International Conference on Networks, Multimedia and Information Technology (NMITCON)*. Bengaluru, India: IEEE, pp. 1–6.

[34] Department of Information Technology VPPCOE&VA, University of Mumbai, Tharkar R. PLANT LEAF DISEASE DETECTION. *INTERANTIONAL J Sci Res Eng Manag* 2024; 08: 1–5.

[35] Saleem MH, Potgieter J, Arif KM. Plant Disease Detection and Classification by Deep Learning. *Plants* 2019; 8: 468.

[36] Applalanaidu MV, Kumaravelan G. A Review of Machine Learning Approaches in Plant Leaf Disease Detection and Classification. In: *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*. Tirunelveli, India: IEEE, pp. 716–724.

[37] Lamba M, Gigras Y, Dhull A. Classification of plant diseases using machine and deep learning. *Open Comput Sci* 2021; 11: 491–508.

[38] Barbedo JGA. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Comput Electron Agric* 2018; 153: 46–53.

[39] Xu M, Kim H, Yang J, et al. Embracing limited and imperfect training datasets: opportunities and challenges in plant disease recognition using deep learning. *Front Plant Sci* 2023; 14: 1225409.

[40] Li L, Zhang S, Wang B. Plant Disease Detection and Classification by Deep Learning—A Review. *IEEE Access* 2021; 9: 56683–56698.

[41] Argüeso D, Picon A, Irusta U, et al. Few-Shot Learning approach for plant disease classification using images taken in the field. *Comput Electron Agric* 2020; 175: 105542.

[42] Atila Ü, Uçar M, Akyol K, et al. Plant leaf disease classification using EfficientNet deep learning model. *Ecol Inform* 2021; 61: 101182.

[43] Xu M, Park J-E, Lee J, et al. Plant disease recognition datasets in the age of deep learning: challenges and opportunities. *Front Plant Sci* 2024; 15: 1452551.

[44] Mohanty SP, Hughes DP, Salathé M. Using Deep Learning for Image-Based Plant Disease Detection. *Front Plant Sci* 2016; 7: 1419.

[45] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. *Commun ACM* 2017; 60: 84–90.

[46] Szegedy C, Wei Liu, Yangqing Jia, et al. Going deeper with convolutions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, pp. 1–9.

[47] PlantVillage Dataset, https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset (accessed 11 March 2025).

[48] Liu B, Zhang Y, He D, et al. Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks. *Symmetry* 2017; 10: 11.

[49] Umar M, Altaf S, Ahmad S, et al. Precision Agriculture Through Deep Learning: Tomato Plant Multiple Diseases Recognition With CNN and Improved YOLOv7. *IEEE Access* 2024; 12: 49167–49183.

[50] Rajamohanan R, Latha BC. An Optimized YOLO v5 Model for Tomato Leaf Disease Classification with Field Dataset. *Eng Technol Appl Sci Res* 2023; 13: 12033–12038.

[51] PlantifyDr Dataset, https://www.kaggle.com/datasets/lavaman151/plantifydr-dataset (accessed 11 March 2025).

[52] Tomato, https://www.kaggle.com/datasets/noulam/tomato (accessed 11 March 2025).

[53] Plant Doc Dataset, https://www.kaggle.com/datasets/abdulhasibuddin/plant-doc-dataset (accessed 11 March 2025).

[54] New Plant Diseases Dataset, https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset (accessed 11 March 2025).

[55] Tomato leaf disease detection, https://www.kaggle.com/datasets/kaustubhb999/tomatoleaf (accessed 11 March 2025).

[56] Baek E-T, Im D-Y. ROS-Based Unmanned Mobile Robot Platform for Agriculture. *Appl Sci* 2022; 12: 4335.

[57] ROS Index, https://index.ros.org/?search_packages=true (accessed 12 June 2025).

[58] Soler Calderé A. *Optimizing agricultural irrigation with autonomous robotic system*. Master Thesis, Universitat Politècnica de Catalunya, https://upcommons.upc.edu/handle/2117/420827 (2024, accessed 12 June 2025).

[59] Sani E <1999>. ROS 2-Based Autonomous Navigation for Agricultural Robots: Sensor Integration and Processing for Traversable Terrain Segmentation. *ROS 2-Based Autonomous Navigation for Agricultural Robots: Sensor Integration and Processing for Traversable Terrain Segmentation*, https://unire.unige.it/handle/123456789/6791 (2023, accessed 12 June 2025).

[60] Tsolakis N, Bechtsis D, Bochtis D. AgROS: A Robot Operating System Based Emulation Tool for Agricultural Robotics. *Agronomy* 2019; 9: 403.

[61] Singh BP, Govindakrishnan PM, Ahmad I, et al. INDO-BLIGHTCAST – a model for forecasting late blight across agroecologies. *Int J Pest Manag* 2016; 62: 360–367.

[62] Afifi MA, Zayan SAM. Implementation of EGY-BLIGHTCAST the first computer simulation model for potato late blight in Egypt.

[63] Landschoot S, De Reu J, Audenaert K, et al. Potentials and Limitations of Existing Forecasting Models for Alternaria on Potatoes: Challenges for Model Improvement. *Potato Res* 2017; 60: 61–76.

[64] Fountas S, Mylonas N, Malounas I, et al. Agricultural Robotics for Field Operations. *Sensors* 2020; 20: 2672.

[65] McGuire M, Soman C, Diers B, et al. High Throughput Soybean Pod-Counting with In-Field Robotic Data Collection and Machine-Vision Based Data Analysis. Epub ahead of print 27 May 2021. DOI: 10.48550/arXiv.2105.10568.

[66] PPO/PRI AGRO Multifunctioneel Landgebruik, Crop and Weed Ecology, PE&RC, et al. *Current potato production in Algeria : an explorative research of the current potato production systems in two regions*. Wageningen: Stichting Wageningen Research, Wageningen Plant Research, Business Unit Plant. Epub ahead of print 2017. DOI: 10.18174/459592.

[67] Soil and temperature conditions for planting potatoes. - Ask Extension, https://ask.extension.org/kb/faq.php?id=860805&utm (accessed 5 June 2025).

[68] growingpotatoes.pdf, https://www.canr.msu.edu/uploads/files/growingpotatoes.pdf?utm_source=chatgpt.com (accessed 5 June 2025).

[69] Oulmane A, Kechar A, Benmihoub A, et al. Assessment of surface water management institutions: a case of public irrigation schemes in northern Algeria. *Water Policy* 2022; 24: 229–241.

[70] Hooker WJ. *Compendium of Potato Diseases*. International Potato Center, 1981.

[71] Van der WJE, Korsten L, Aveling T a. S. A review of early blight of potato. *Afr Plant Prot* 2001; 7: 91–102.

[72] Runno-Paurson E, Loit ,Kaire, Hansen ,Merili, et al. Early blight destroys potato foliage in the northern Baltic region. *Acta Agric Scand Sect B — Soil Plant Sci* 2015; 65: 422–432.

[73] VARMAl PK, Gandhp SK, Singh S. Biological control of Alternaria solani, the causal agent of early blight of tomato.

[74] Arora RK, Sharma S, Singh BP. LATE BLIGHT DISEASE OF POTATO AND ITS MANAGEMENT. *Potato J*; 41, https://epubs.icar.org.in/index.php/PotatoJ/article/view/41808 (2014, accessed 5 June 2025).

[75] Irish Famine. In: *Encyclopedia of World Poverty*. 2455 Teller Road, Thousand Oaks California 91320 United States: Sage Publications, Inc. Epub ahead of print 2006. DOI: 10.4135/9781412939607.n369.

[76] Forbes GA, Landeo JA. Late Blight. In: *Handbook of Potato Production, Improvement, and Postharvest Management*. CRC Press, 2006.

[77] Late blight of tomato and potato, https://extension.umn.edu/disease-management/late-blight (accessed 5 June 2025).

[78] Gevens A, Wilbur J. Potato Late Blight: Identification and Management (A4052-02).

[79] Slininger PJ, Schisler DA, Ericsson LD, et al. Biological control of post-harvest late blight of potatoes. *Biocontrol Sci Technol* 2007; 17: 647–663.

[80] Yildiz M. *Potato: From Incas to All Over the World*. BoD – Books on Demand, 2018.

[81] Tumwine J, Frinking HD, Jeger MJ. Integrating cultural control methods for tomato late blight (*Phytophthora infestans*) in Uganda. *Ann Appl Biol* 2002; 141: 225–236.

[82] Weather API - OpenWeatherMap, https://openweathermap.org/api (accessed 10 June 2025).

[83] Free Open-Source Weather API | Open-Meteo.com, https://open-meteo.com/ (accessed 10 June 2025).

[84] Musah A, Dutra LMM, Aldosery A, et al. An Evaluation of the OpenWeatherMap API versus INMET Using Weather Data from Two Brazilian Cities: Recife and Campina Grande. *Data* 2022; 7: 106.

[85] Khobor O, Senyk A, Lytvyn V. USAGE OF INFORMATION TECHNOLOGIES TO PREVENT METEOROLOGICAL RISKS. *Her Khmelnytskyi Natl Univ Tech Sci* 2023; 323: 332–336.

[86] Romero Rodríguez L, Guerrero Delgado Mc, Castro Medina D, et al. Forecasting urban temperatures through crowdsourced data from Citizen Weather Stations. *Urban Clim* 2024; 56: 102021.

[87] Evcin E, Erten YM. Comparison of Different Weather Data Acquisition Methods. In: *2024 9th International Conference on Computer Science and Engineering (UBMK)*. Antalya, Turkiye: IEEE, pp. 1–6.

[88] Silva Filho JB, Fontes PCR, Ferreira JFDS, et al. Best Morpho-Physiological Parameters to Characterize Seed-Potato Plant Growth under Aeroponics: A Pilot Study. *Agronomy* 2024; 14: 517.

[89] Dagne Z, Dechassa N, Mohammed W. Influence of Plant Spacing and Seed Tuber Size on Yield and Quality of Potato (Solanum tuberosum L.) in Central Ethiopia. *Adv Crop Sci Technol*; 06. Epub ahead of print 2018. DOI: 10.4172/2329-8863.1000406.

[90] (PDF) Robustless Test Of Several Potato Clon In The Medium Plains. *ResearchGate*. DOI: 10.9790/2380-1008016165.

[91] Effective Methods for Potato Spacing | Properly Rooted, https://properlyrooted.com/potato-spacing/ (2023, accessed 10 June 2025).

[92] Kacheyo OC, Schneider HM, De Vries ME, et al. Shoot Growth Parameters of Potato Seedlings are Determined by Light and Temperature Conditions. *Potato Res* 2024; 67: 1159–1186.

[93] Björkner B, Frick-Engfeldt M, Pontén A, et al. Plastic Materials. In: Johansen JD, Frosch PJ, Lepoittevin J-P (eds) *Contact Dermatitis*. Berlin, Heidelberg: Springer, pp. 695–728.

[94] Krasnyi BL, Tarasovskii VP, Rakhmanova EV, et al. Chemical Resistance of Ceramic Materials in Acids and Alkalis. *Glass Ceram* 2004; 61: 337–339.