UNIVERSITÄT BONN

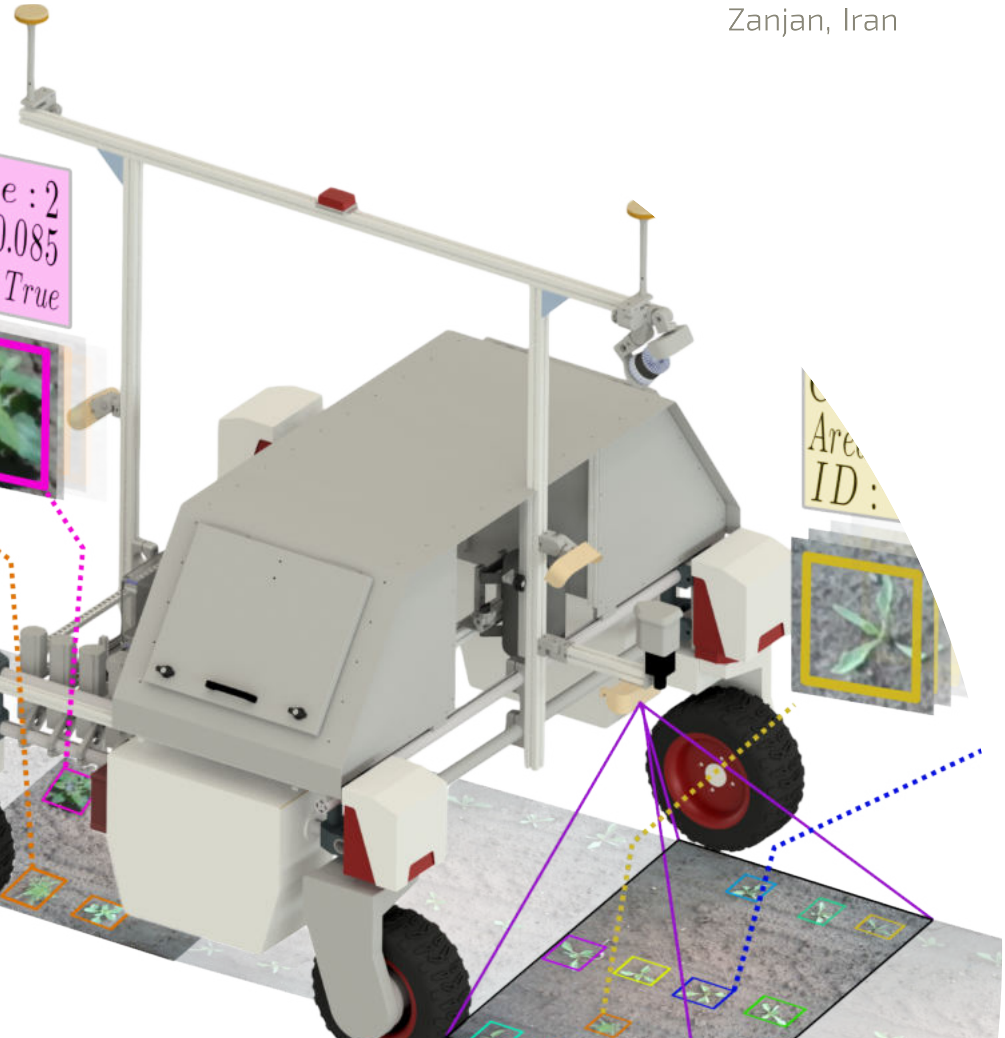
Inaugural-Dissertation
zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Agrar-, Ernährungs- und Ingenieurwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn
Institut für Landtechnik

Precision Weed Management Enabled by Robotic and Robotics Vision

von

Alireza Ahmadi

aus
Zanjan, Iran



Bonn 2025

Referent:
Prof. Dr. Chris McCool, Universität Bonn, Deutschland

1. Korreferent:
Prof. Dr. Maren Bennewitz, Universität Bonn, Deutschland

2. Korreferent:
Prof. Dr. Marc Hanheide, University of Lincoln, England

Tag der mündlichen Prüfung: 28.04.2025

Erscheinungsjahr: 2025

Angefertigt mit Genehmigung der Agrar-, Ernährungs- und Ingenieurwissenschaftlichen Fakultät der Universität Bonn

Erklärung der Urheberschaft

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenenen Hilfsmittel angefertigt habe; die aus fremdem Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Ort, Datum

(Unterschrift)

Zusammenfassung

In den letzten Jahrzehnten hat sich das traditionelle Unkraut- und Pflanzenmanagement stark auf Herbizide und mechanische Unkrautbekämpfung verlassen. Diese Methoden haben erhebliche ökologische und landwirtschaftliche Herausforderungen mit sich gebracht. Weltweit werden jährlich über 2 Millionen Tonnen Herbizide eingesetzt, was Bedenken hinsichtlich der Lebensmittelsicherheit, Umweltschäden und Gesundheitsrisiken für den Menschen aufwirft. Die Resistenz von Unkräutern gegen Herbizide ist ein wachsendes Problem, mit über 500 gemeldeten Fällen weltweit. Gleichzeitig drängen Verbraucher auf organische, chemiefreie Lebensmittel, was Landwirte dazu zwingt, den Einsatz von Agrochemikalien zu reduzieren und gleichzeitig hohe Erträge aufrechtzuerhalten. Diese Situation verdeutlicht die dringende Notwendigkeit innovativer, nachhaltiger Lösungen für die Landwirtschaft.

Diese Dissertation untersucht Technologien der Präzisionslandwirtschaft, mit einem besonderen Fokus auf biodiversitätsbewusste robotische Systeme für pflanzenindividuelle Unkrautbekämpfung auf Ackerflächen. Wir haben versucht, die Einschränkungen konventioneller Unkrautmanagementmethoden zu überwinden, indem wir fortschrittliche robotische Lösungen unter Einsatz von maschinellem Sehen, Deep Learning und autonomer Navigation für nachhaltige und gezielte Interventionen in realen Anwendungen vorschlagen. Die zentrale Innovation konzentriert sich auf die Entwicklung einer neuartigen Plattform für Präzisionsunkrautbekämpfung und Pflanzenüberwachung namens BonnBot-I. Diese Plattform ist mit fortschrittlichen Sensoren und Rechenwerkzeugen ausgestattet, um autonome Operationen in verschiedenen Ackerbauumgebungen durchzuführen.

Ein Hauptthema der landwirtschaftlichen Autonomie ist die zuverlässige autonome Navigation in unübersichtlichen landwirtschaftlichen Umgebungen mit eingeschränkter globaler Lokalisierung, wie GPS. Angesichts der Tatsache, dass immer noch ein groSSer Teil der Ackerflächen nicht mit GPS-gestützten Systemen eingesät wird, könnte die Integration von Navigationsmethoden, die auf lokalen Beobachtungen basieren, dazu beitragen, Umweltprobleme zu lösen, die zuverlässige Navigation ermöglichen und Pflanzenschäden minimieren. Daher stellen wir einen visionbasierten Navigationsansatz vor, der den BonnBot-Idurch Pflanzenreihen mit unterschiedlichen Kronentypen und Kultivaren allein auf Basis von Echtzeitkameradaten führt.

Ein zentrales Ziel dieser Dissertation ist es, ein robustes Framework für die Entwicklung von Robotern zu etablieren, die in der Lage sind, präzise, pflanzenindividuelle Unkraut- und Pflanzenpflege auf Ackerflächen mit einer Vielzahl von Kultivaren und Unkrautdichten durchzuführen. Dafür ist ein genaues Überwachungssystem für Pflanzen und Unkraut erforderlich, um Unkrautbekämpfungsstrategien auf Grundlage der vorhandenen Pflanzeninstanzen zu gestalten. Um dieses Ziel zu erreichen, integriert BonnBot-Ifortschrittliche instanzbasierte semantische Segmentierungs- und Tracking-via-Segmentierungs-Methoden. Unser Ansatz ermöglicht es, einzelne Pflanzen in Echtzeit zu identifizieren und zu verfolgen, sie nach Art, GröSSe, Wachstumsstadium und genauer Position unter realen Feldbedingungen zu kategorisieren. Diese fortschrittlichen Systeme erlauben es, umweltfreundliche Unkrautbekämpfungsstrategien umzusetzen, die auf spezifische Pflanzen in realen landwirtschaftlichen Umgebungen zugeschnitten sind. Diese Innovation ermöglicht die Priorisierung auf Pflanzenebene und die Durchführung gezielter Interventionen basierend auf den individuellen Bedürfnissen jeder Pflanze mithilfe des neuartigen Unkrautbekämpfungswerkzeugs von BonnBot-I. BonnBot-Iist mit einem spezialisierten Unkrautbekämpfungswerkzeug ausgestattet, das unabhängig steuerbare Linearschienen und Sprühdüsen umfasst, um diese selektiven Interventionen zu erleichtern. Dieses Design ermöglicht es BonnBot-I, hochpräzise Anwendungen durchzuführen, den Bedarf an Agrochemikalien erheblich zu reduzieren und die mit herkömmlichen Methoden verbundenen Umweltauswirkungen zu minimieren.

AbschlieSSend zeigt diese Dissertation, wie Robotik und künstliche Intelligenz (KI) die Zukunft des Pflanzenmanagements durch innovative biodiversitätsbewusste und pflanzenindividuelle Unkrautbekämpfungspraktiken grundlegend verändern können. Durch die Integration fortschrittlicher maschineller Bildverarbeitung, Deep Learning und autonomer Navigation bietet BonnBot-Ieinen einzigartigen Ansatz für nachhaltige Landwirtschaft, der die Biodiversität respektiert und die Umweltgesundheit priorisiert. Im Gegensatz zu herkömmlichen Unkrautbekämpfungsmethoden, die auf eine einheitliche Anwendung von Herbiziden oder mechanische Entfernung setzen und oft umliegende Pflanzen und Ökosysteme schädigen, bietet BonnBot-Ipräzise Eingriffe, die individuell auf einzelne Pflanzen abgestimmt sind.

Abstract

In recent decades, traditional crop and weed management has heavily relied on herbicides and mechanical weeding. These methods have caused significant environmental and agricultural challenges. Over 2 million tons of herbicides are used annually globally, raising concerns about food safety, environmental harm, and human health risks. Weed resistance to herbicides is a growing problem, with over 500 cases reported worldwide. Meanwhile, consumer demand for organic, chemical-free food pushes farmers to reduce agrochemical use while maintaining high yields. This situation highlights the urgent need for innovative, sustainable farming solutions.

This thesis explores precision agriculture technologies, focusing on biodiversity-aware robotic systems for plant-level weeding in arable farms. We tried to address the limitations of conventional weed management, by proposing advanced robotic solutions using machine vision, deep learning, and autonomous navigation for sustainable and targeted interventions in the real world. The core innovation is centered on developing a novel precision weeding and crop-monitoring robot platform called BonnBot-I. This platform is equipped with advanced sensors and computational tools to conduct autonomous operations in diverse arable farming environments.

One of the main topics in agricultural autonomy is performing reliable autonomous navigation in cluttered farming environments with poor global localization accessibility like GPS. Considering the fact that still a large portion of the arable farms are not seeded using GPS-guided systems, integration of local observations-based navigation methods could relieve environmentally posed challenges for robots to achieve reliable navigation and minimize crop damage. Hence, we introduce a vision-based navigation approach that guides the BonnBot-I through rows of crops with different canopy types and cultivars relying only on the real-time camera data.

A central aim of this thesis is to establish a robust framework for developing robots capable of conducting precise, plant-specific weed and crop management in arable farms that feature a variety of cultivars and weed densities. Hence an accurate crop and weed monitoring system is needed to shape weeding strategies based on the presence of plant instances. To fulfill this requirement, BonnBot-I incorporates cutting-edge instance-based semantic segmentation and tracking-via-segmentation methods. Our approach enables the identification and tracking of individual plants in real time, categorizing them by species, size, growth stage, and precise location under actual field conditions. These advanced systems allow us to implement eco-friendly weeding strategies tailored to specific plants in real agricultural settings. This innovation enables plant-level prioritization and the execution of targeted interventions based on each plants unique needs using BonnBot-I's novel weeding tool. BonnBot-Iis equipped with a specialized weeding tool, including independently controllable linear axes and spray nozzles, facilitating these selective interventions. This design enables BonnBot-I to perform highly precise applications, significantly reducing the need for agrochemicals and minimizing the environmental impact associated with traditional broadcast methods.

In conclusion, this thesis demonstrates how robotics and artificial intelligence (AI) can profoundly reshape the future of crop management through innovative biodiversity-aware and plant-specific weeding practices. By integrating advanced machine vision, deep learning, and autonomous navigation, BonnBot-I provides a unique approach to sustainable agriculture that respects biodiversity and prioritizes environmental health. Unlike traditional weeding methods that rely on uniform herbicide application or mechanical removal, which often harm surrounding crops and ecosystems, BonnBot-I offers precision interventions tailored to individual plants.

Acknowledgements

My journey in agricultural robotics began in 2017, during my masters studies, when I took on a student job in the IPB Lab at the University of Bonn. There I had the opportunity to design and build an agricultural robot, marking my first experience in Agricultural and field robotics. At the time, I was new to the field, but quickly realized how unexplored and rich in opportunity it was. Having worked for nearly a decade with indoor robotic systems, the challenge of developing systems for outdoor, agricultural environments brought me immense fulfillment. The potential for my work to make a tangible impact in agriculturea sector with such direct and profound effects on peoples liveswas both motivating and humbling.

Now after almost 8 years completing my Ph.D. at the AgRobotics Lab in the Institute für Landtechnik at the University of Bonn has been an enriching experience. This role has allowed me to dive deeply into areas Ive long been passionate about and has granted me the privilege to work on projects with real-world applications. I am immensely grateful to have contributed to a field that has the potential to have positive and lasting impacts on food production and sustainability. This purpose has kept me moving forward, even through challenges like broken robots, code that refused to compile, and models that often needed rethinking.

My sincere thanks go first to my advisor, Chris McCool, who has been a constant source of support, mentorship, and inspiration. His willingness to share knowledge, engage in lengthy discussions and provide constructive feedback has been invaluable in my development as a researcher and engineer. I am particularly grateful for his encouragement to pursue my ideas independently, as well as for his empathy and understanding, which allowed me to pursue my interests in making a startup out of my Ph.D. project while I was still doing my Ph.D.

I especially thank Anke Indorf, whose expertise in navigating German bureaucracy made my life in Bonn much smoother. The authors also express their gratitude to their colleagues in the AgRobotics Lab. Working with Claus Smitt from day one has been a joy his patience and support have been invaluable. Michael Halstead has been a steady source of help, especially as deadlines were getting closer, and I thank him for his dedication. Patrick Zimmer brought our team together with his energy and the many barbecues and social gatherings he organized. I am also grateful to Maohui Li, and Esra Guclu, whose shared insights, collaboration, and camaraderie have enriched my experience.

I am also grateful for the support from the Phenorob project and colleagues in the Phenorob office. The Phenorob office has been a source of continuous assistance and support, making my work productive and enjoyable. The collaborations and friendships formed within this wider community have been incredibly valuable, and I have cherished the gatherings and events that have made my Ph.D. journey more memorable. Special thanks to Julio Pastrana, CEO of Escarda Technologies Gmbh, who first encouraged me to apply to the AgRobotics Lab and has been a constant source of support and giving me a chance to contribute to his startup as a junior developer. I am especially grateful to Dr. Shiri Ghidari, who first gave me the opportunity to work in the robotics lab of the Polytechnic University of Tehran, and to my dear friends Mojtaba Karimi, Mohsen Fallahi, Omid Jahanian, Edwin barbarians, and Poria Younes-Sotoudeh, whose encouragement, guidance, friendship, and mentorship were invaluable in my journey of life.

Finally, I owe everything to my family, especially my parents, who instilled in me the confidence to follow my passions. Above all, I thank my wife, Faezeh, whose unwavering support and encouragement made this journey possible. Her sacrifices, patience, and love have been my greatest source of strength and I am deeply grateful for your companionship.

Thank you.

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germanys Excellence Strategy - EXC 2070 390732324.

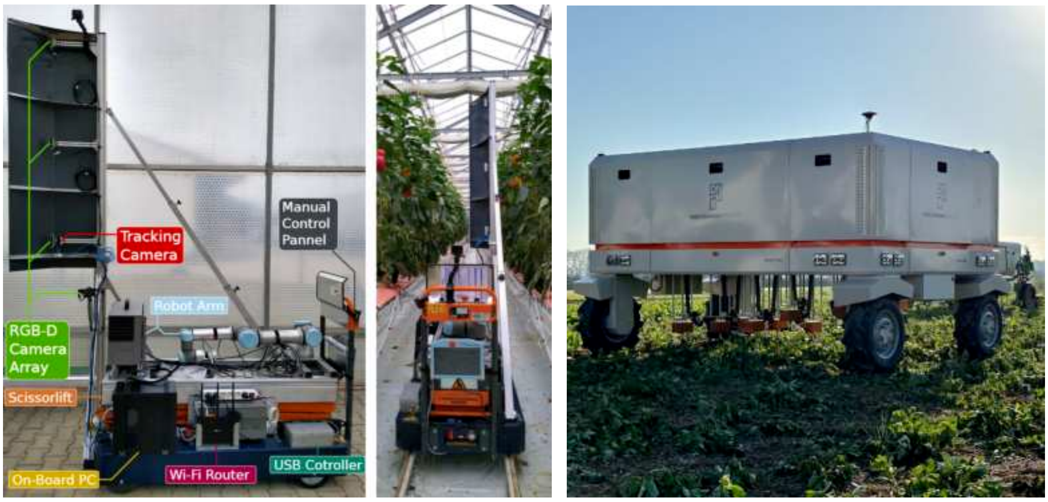
Contents

Chapter 1

Introduction

Conventional crop and weed management methods predominantly depend on uniform broadcast approaches like herbicides and mechanical implements, contributing to significant issues within agri-food systems globally [74]. Annually, the world sees the use of approximately 2 million tons of herbicides, leading to widespread concerns over food safety, persistent environmental damage, and considerable risks to both ecosystems and human health. The problem of herbicide resistance is getting more crucial. Currently, over 500 cases of herbicide resistance have been reported worldwide. This involves 167 different herbicides and approximately 300 weed species and these numbers continue to grow rapidly [63, 127, 109].

Along with herbicide residences, the agricultural industry is being pushed to reduce agrochemicals due to the social desire to consume more natural and organic



(a) PathoBot from AgRobotics Uni Bonn

(b) Robot One from Pixel Farming Robotics

Figure 1.1: Agricultural Platforms; (a) PATHoBot with sensors, arm, and other components highlighted and operated in the glasshouse, (b) Robot One from Pixel Robotics Company.

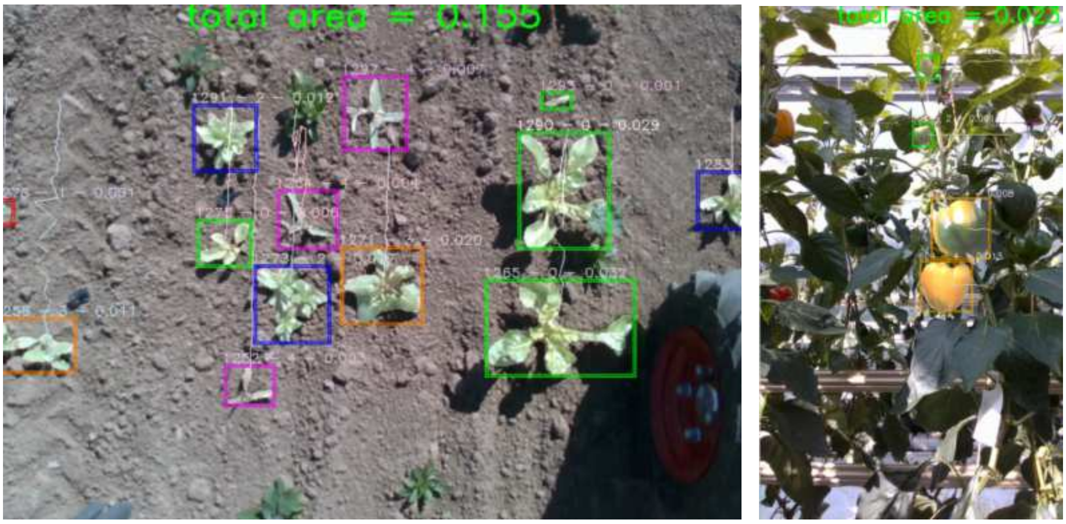
foods. [23, 127]. Hence, society and governments encourage farmers to promote the biodiversity of their fields. To enable this, “smart farming techniques” need to address the challenges ranging from automated navigation [3], crop monitoring [100, 52], and through to management actions such as weeding [80].

Such technologies have advanced rapidly in the context of agricultural robotics, enabling precise, plant-level interventions that align with social pressures for sustainable farming, such as reducing herbicide use and promoting bio-based food production. These technologies address farmers’ needs for more efficient operations by integrating diverse tools, including physical, chemical, and laser-based methods [27, 119, 107]. Given the diversity of crops, weeds, and field conditions, adaptable robotic systems are essential to ensure effective and environmentally conscious farming operations [6, 72].

The ever-changing environment, including varying weather conditions, different soil types, and the growth stages of crops and weeds complicate reliable navigation in arable fields. To enable the smooth integration of smart farming technologies in farmers’ everyday practices, we need to ensure that agricultural vehicles or robots can traverse the field flawlessly at the desired speed and accuracy [7]. A key component of this is accurate and reliable navigation, which will reduce crop damage from the wheels. To achieve this, platforms need to be robust to technology outage,s meaning multiple approaches to navigation are required. One potential solution to the navigation problem is using the Global Navigation Satellite System (GNSS). Such an approach has been used for both agricultural machinery [109] and robotic platforms [19]. The downside of this approach is that it relies on an expensive sensor and suffers from limitations such as possible GNSS outages and reliance on geo-referenced auto-seeding. Thus, crop-based navigation techniques leveraging the field structure were investigated for autonomous guidance [21, 7] and in-field interventions [12].

Crop monitoring is essential across all types of farming, from arable crops like wheat, corn, and sugar beet to horticultural crops such as apples, tomatoes, and sweet peppers [52]. In arable farming, weed presence, type, and density are critical factors, and various platforms have been developed specifically to manage them [16, 78]. Furthermore, effective plant-level interventions need to operate in different fields with varying crops, weed species, and weed distributions. To achieve sustainable, human-level weeding, it is crucial to accurately identify weed species and precisely locate targets, enabling strategic decision-making and targeted interventions. [22, 82].

Weeding is crucial in farming as weeds compete with crops for soil nutrients, potentially lowering yields [97]. In recent years, robotic weed management techniques have rapidly evolved, intending to treat each weed as precisely as possible. This will reduce the amount of agrochemicals used and minimize the environ-



(a) Sugar-beet.

(b) Sweet pepper.

Figure 1.2: Monitoring outputs; monitoring algorithms provide real-time crop insights, including ripeness, species identification, and area estimation, to support informed decisions on weeding, harvesting, and yield estimation.

mental side-effects of the weeding intervention method [42]. Robotic weed intervention has the potential to revolutionize weeding paradigms through plant-level weed management.

Furthermore, current broadcast mechanical hoeing technology could, in some situations, lead to soil erosion and ecosystem disturbances [46]. Additionally, research has shown that broadcast hoeing can heavily impact ecosystems by speeding up soil nutrient loss, and water evaporation, and encouraging new weeds due to rotation of soil. Such indiscriminate application, affecting both crops and weeds, not only undermines the health of beneficial organisms but also accelerates biodiversity loss [46, 127]. Traditionally, farming has relied on manual weeding Figure 1.3, however, this approach is labor-intensive, time-consuming, and expensive. These negatives make this approach prohibitive in today's financial farming climate. Hence, there is a need for technology to enable farmers to manage weeds in their farms with the least environmental effect, high efficiency, and comparable costs. We believe the research presented in this thesis could counteract such failing trends with the least invasive and targeted intervention approach.

Several robotic weed control platforms have been introduced offering active and passive interventions in the field. Considering the need to operate in different fields with varying crops, weed species, and weed distributions, it is clear that there is no one best solution. These robotic platforms should also be able to cater to a variety of tools [72, 6], and several approaches have been proposed to accommodate these different tools [27, 72, 119, 10, 121]. The concept of plant-

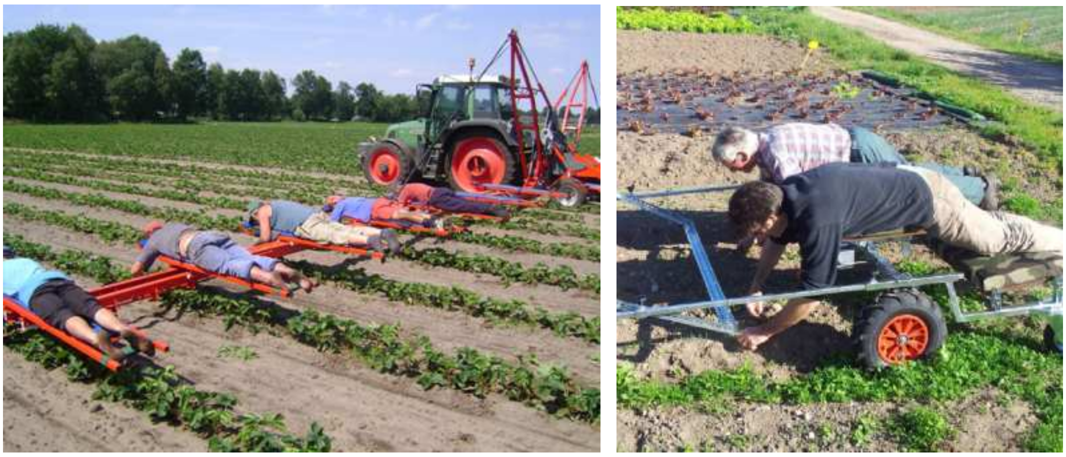
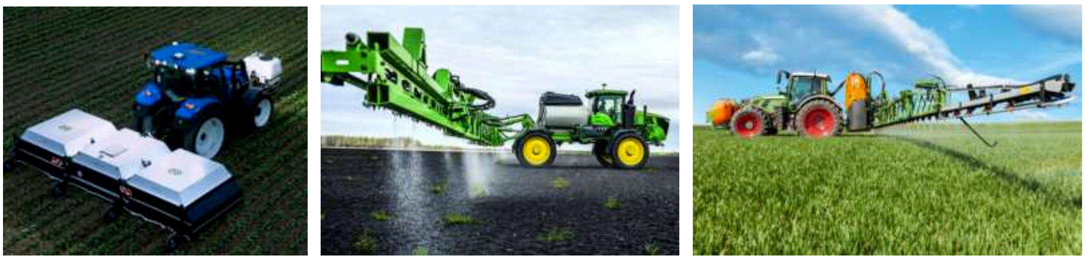


Figure 1.3: Organic manual weeding solution; a time-consuming, laborious, and costly method that needs almost 200 hours per hectare.

specific treatment is referred to as bio-diversity-aware interventions [127], The current state-of-the-art agricultural weeding technology spans from traditional hand-picking and mechanical hoeing machines to chemical herbicide sprayers and nascent robotic weeders. These methods present significant challenges, particularly due to the environmental impact associated with heavy dependence on any single method. Such uniform and widespread treatment strategies have been shown to contribute to adverse ecological trends, including the development of resistance. These approaches can result in soil degradation, water contamination, nutrient loss, and a reduction in biodiversity. However, some of the more advanced systems now employ visual recognition technology to differentiate between crops and weeds, enabling precise chemical applications and mechanical hoeing between rows and crops. Below, we will outline the key technologies in precision weeding, discussing their advantages and disadvantages.

Spot-Spray: The Spot-Spray systems perform plant-level intervention by differentiating between crop and weed. While chemical applications are the most effective approach in today's agriculture, their suitability is restricted by factors like precision and environmental conditions. Moreover, the uniform use of chemical herbicides encourages the emergence of weed species resistant to these chemicals, diminishing the efficacy of these methods over time [36, 8]. As an illustration, a prominent Spot-Spray system on the market is priced at approximately 150,000 euros. It offers a maximum precision of 8cm for spray areas and can operate at speeds up to 5km/h. Nonetheless, these systems often lack the versatility to accommodate various crop types and growth stages due to relying only on one herbicide type [89].

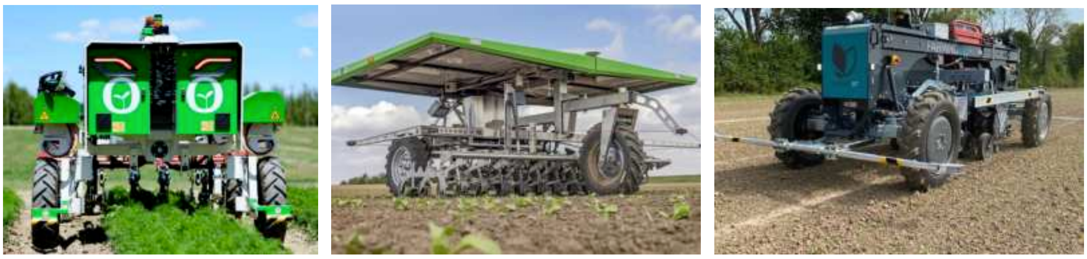
Mechanical Inter-row Hoeing: Recent mechanical hoeing systems introduce an advanced inter- and intra-row hoeing mechanism for row-crop fields. These systems employ a moving fork with constant contact with soil that is par-



(a) Eco-Robotics - ARA [89] (b) John Deere; See and Spray [36] (c) Spot Spray - Amazone [8]

Figure 1.4: Spot-Spray weeding system; (a) ARA, (b) See and Spray, and (c) Amazone are advanced spot-spray weeding systems designed to precisely target and eliminate weeds using real-time detection precise herbicide application. These technologies reduce chemical usage while enhancing the accuracy and efficiency of weed management in agriculture.

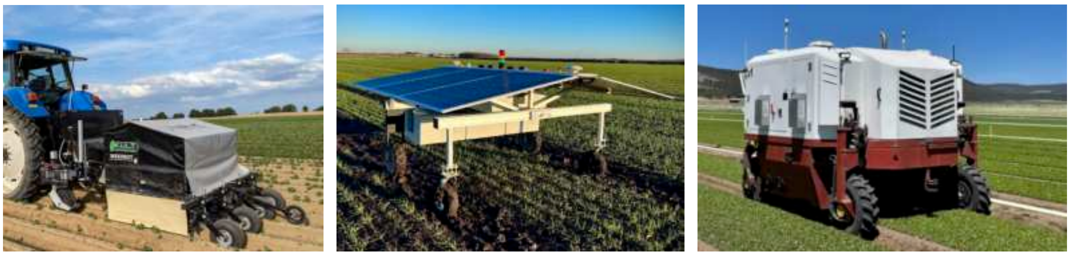
ticularly used in organic farming, eliminating the need for herbicides [108, 44, 86]. However, it carries the risk of harming crops, especially in densely planted areas. Its efficacy diminishes in areas with a high density of weeds, and the disruption caused to the soil's structure may adversely affect soil health. Additionally, overuse of this method can lead to the emergence of new weeds and contribute to soil erosion. Moreover, this widespread disturbance of the soil threatens the fertility of cultivation lands by accelerating the loss of nutrients.



(a) Naio Technologies [108] (b) Farm Droid [44] (c) Farming GT [86]

Figure 1.5: Mechanical Hoeing weeding systems; (a) Naio, (b) Farm droid, and (c) Farming GT from farming revolution is leading mechanical hoeing systems that use robotics to remove weeds through mechanical hoeing systems autonomously. These systems provide a chemical-free solution for weed control, promoting sustainable agriculture while reducing labor and herbicide reliance.

Thermal Applications (Laser-based or Electricity-based): Thermal weeding technologies represent some of the latest advancements within the selective intervention category [115, 39, 87]. Laser-based weeding stands out due to its efficacy. Some of the limitations associated with this method of intervention include strict safety standards, high upfront costs, and significant energy demands. These limitations contribute to its expense and question the sustainability of future farming. Operating these complex systems necessitates expert knowledge, and their utility can be hindered by unfavorable weather conditions, affecting their adaptability to various farming environments, including different growth stages and levels of air and soil moisture. Specifically, the timing and growth phase of the weeds is critical for this technology's success; larger weeds diminish its efficiency substantially, demanding more energy and time for effective management.



(a) WeedBot [115]

(b) Earth-Rover [39]

(c) Carbon Robotics [87]

Figure 1.6: Thermal weeding systems; these images showcase thermal weeding systems from (a) WeedBot, (b) Earth-Rover, and (c) Carbon Robotics, which use robotic technology for autonomous weed removal through burning mostly early-stage weeds. These systems offer an eco-friendly alternative to chemical weeding and less environment-disturbing means to manage weeds, making them highly effective in sustainable agriculture.

When considering the challenges associated with available weeding methods, it becomes evident that each technique, when applied uniformly across all plant types, introduces its own set of issues. Spot-spraying can lead to excessive chemical use and environmental contamination, and developing of new herbicide resistance. mechanical hoeing can disrupt soil structure and harm beneficial organisms. However, by integrating these methods into a single, adaptive system that can dynamically choose the appropriate technique at the appropriate time would significantly enhance both weeding efficacy and biodiversity. This targeted approach minimizes the drawbacks of each method, leveraging their strengths to effectively manage weeds while preserving the health of the crop and the surrounding ecosystem. By doing so, we not only optimize agricultural productivity but also contribute to a more sustainable and ecologically responsible farming practice. The treatment of each plant must be dictated by its species, size, and impact upon not only the crop but also the environment [22, 52]. One of the key elements to achieve precise weeding is plant-level treatment, where the treatment of each plant is dictated by its species, size, and its impact upon not only the crop but also the environment [22]. These approaches rely on the underlying perception or agricultural monitoring approaches which have gained significant research attention in recent years, including in glasshouses [101, 99], orchards [110], and fields (for weed intervention) [78].

In response, we developed a novel weeding and crop management system that can be mounted on any weeding trailer (or tractor) or carried with an autonomous robot. Leveraging instance-based semantic segmentation and tracking-via-detection methods based on deep neural networks (DNNs), our prototype distinguishes each plant by its growth stage, type, and potential harm to crops. Hence, deciding on the most effective intervention method to deal with individual plants on the farms. Furthermore, we propose a unique multi-head weeding tool that permits plant-level selective treatments. The proposed technology can streamline the integration and reduce the costs for the stakeholders. This architecture not only facilitates targeted interventions for individual plants and encourages biodiversity-oriented interventions but also supports operation across a diverse range of plant species and environmental conditions, therefore significantly lowering barriers to adoption and deployment. The journey toward fully autonomous precision agriculture faces significant challenges. This thesis addresses four main questions critical to advancing this field:

1. What design features are essential for creating a robust, farming robot capable of crop monitoring, plant-level intervention, and autonomous navigation?
2. How can a farming robot achieve reliable autonomous navigation using only local and visual (camera-based) observations in row-crop farms?
3. How can we enable a weeding robot to perform crop-agnostic, plant-level weeding while considering biodiversity-aware practices?
4. Can a robotic system learn to conduct precision weed management based on diverse plant characteristics and priorities provided by experts?

1.1 Main Contributions

This thesis investigates the application of robotic vision to automate precision weed management and crop monitoring in arable farming. The work focuses on developing advanced methods that leverage both classical computer vision (CV) and deep learning techniques to enhance the autonomy and effectiveness of agricultural robots. By integrating these technologies, this thesis demonstrates how robotic systems can significantly improve field navigation, weed control, and sustainable crop management, pushing the boundaries of what is achievable in precision agriculture.

We begin by introducing our novel precision crop/weed management and monitoring robot called BonnBot-I. This is followed by addressing the fundamental challenges associated with autonomous navigation in arable fields, where robots must operate reliably in dynamic and often unpredictable environments, and present our solutions using classical machine vision and DNN-based approaches. Additionally, we explore the capabilities of BonnBot-I in dealing withthe complexities of plant-level intervention, particularly in the context of biodiversity-aware practices that prioritize environmental sustainability practices. The ultimate goal of this work is to provide a comprehensive framework for developing and deploying agricultural robots that can autonomously perform precise and sustainable interventions in arable farms.

This thesis is organized into six chapters, each addressing a specific aspect of autonomous precision agriculture, with a focus on developing robotic vision and precision framing capabilities.

Chapter 2 is dedicated to explaining the fundamental techniques that underpin our proposed precision agriculture systems. These foundational methods form the building blocks for autonomous precision weed management and crop monitoring solutions. By reviewing these core principles, we establish a basis for understanding how our systems effectively handle tasks such as crop surveying, weed detection, and field navigation that are introduced in the following contribution chapters.

We introduce BonnBot-I in Chapter 3, a precision weeding and crop monitoring robot designed to address the first research question. BonnBot-I is equipped with advanced sensors, powerful computation, and a unique intervention tool that allows it to perform real-time, precise surveillance and intervention in farming fields. Section 3.1 introduces different important sections of the platform and the ideology behind their design. This continues with Section 3.2, where we introduce simulation modalities developed especially for this platform with focus on precision weeding applications. Section 3.3 demonstrates three specially designed and publicly available datasets created and used in the projects enabled by BonnBot-I. These data were captured over three years at the Campus Klein-Altendorf of the University of Bonn using BonnBot-I. These datasets are utilized extensively to evaluate the performance of the techniques developed throughout this thesis. Finally, Section 3.4, explores the techniques used to enhance the performance of DNN-based monitoring systems by leveraging geometric and motion information captured by BonnBot-I. The capabilities of this robotic platform are critical to the success of the interventions and operations discussed in later chapters.

Chapter 4 is dedicated to the topic of autonomy in farming sites, and it demonstrates our vision-based approaches for in-field navigation. This approach relies solely on camera inputs to guide the robot through the crop rows in arable farms. Initially, Section 4.1 reviews the existing research on autonomous navigation in row-crop fields and agricultural robotics and their pros and cons. We then present our proposed strategy and approach for in-field autonomous navigation using only local, onboard observations in Section 4.2. Next, Section 4.6 details the experimental evaluations and implementation insights from both simulated and real-field tests. Finally, conclusions and future directions are discussed in Section 4.7.

Chapter 5 is dedicated to the third research question of this thesis; addressing plant-level bio-diversity-aware weeding using robotics and robotics vision. We begin by reviewing similar robotic weeding approaches aimed at precision farming. Next, in Section 5.2, we define a standard weeding scenario and outline the system requirements for using BonnBot-I. In Section 5.3, we introduce our method for modeling real-field observations into a usable format for weeding. We then discuss a target-space management strategy in Section 5.4 for real-time interventions. In Section 5.5, we explain the path-planning strategies that control BonnBot-I's intervention heads, leading to the introduction of a biodiversity-aware weeding scheme in Section 5.6. Finally, Section 5.7 details our experiments in both simulated and real-world settings, assessing the performance of our approach in sugar beet and corn fields. The chapter concludes with a discussion on the design, deployment, and potential future improvements of this weeding system.

The final chapter focuses on the application of reinforcement learning (RL) techniques to optimize plant-level interventions. In Chapter 6 we leverage RL to perform plant-level in-field interventions considering multi-modal observation using the novel weeding tool of BonnBot-I. After a comprehensive introduction and review of the literature, we introduce the concept of the RL framework developed for BonnBot-Iin Section 6.2, which enables the robot to learn and adapt its intervention strategies based on plant priorities. In the follow-up sections, topics of observation modeling Section 6.3, action space deign Section 6.4 and reward function Section 6.5 are discussed. Finally, in Section 6.6 we present experimental results and their direct comparison to the SOTA methods presented in Section 6.7, showing the advantages and limitations of the RL-based approach.

This thesis illustrates the transformative potential of integrating robotic vision and advanced machine learning techniques to enhance the autonomy, precision, and sustainability of agricultural robots. By deploying these cutting-edge technologies in systems like BonnBot-I we not only improve the accuracy and efficiency of field interventions but also promote a more sustainable approach to farming. These advancements have the potential to fundamentally change the way we manage and monitor crops, leading to more resilient agricultural practices and paving the way for a new era of sustainable, tech-driven farming. Through this work, we demonstrate that the future of agriculture lies in the seamless integration of intelligent robotics with traditional farming techniques, ultimately contributing to more sustainable food production systems worldwide.

1.2 Publications

Parts of this thesis have been published in the following peer-reviewed conference and journal articles, for which I have been the main contributor:

- **Alireza Ahmadi**, Michael Halstead, and Chris McCool. Virtual Temporal Samples for Recurrent Neural Networks: applied to semantic segmentation in agriculture [4]. *German Conference on Pattern Recognition (DAGM), 2021.*
  DOI: 10.1007/978-3-030-92659-537.

- **Alireza Ahmadi**, Michael Halstead, and Chris McCool. Towards Autonomous Visual Navigation in Arable Fields [3]. *IEEE Intelligent Robots and Systems conference (IROS), 2022.*
  DOI: 10.1109/IROS47612.2022.9981299.

- **Alireza Ahmadi**, Michael Halstead, and Chris McCool. BonnBot-I: A Precise Weed Management and Crop Monitoring Platform [5]. *IEEE Intelligent Robots and Systems conference (IROS), 2022.*
  DOI: 10.1109/IROS47612.2022.9981304.

- **Alireza Ahmadi**, Michael Halstead, Smitt, Claus, and Chris McCool. BonnBot-I Plus: A Bio-diversity Aware Precise Weed Management Robotic Platform [6]. *IEEE Robotics and Automation Letters, 9(7):6560-6567, June 2024 Presented in IEEE International Conference on Robotics and Automation - ICRA40, 2024.*
  DOI: 10.1109/LRA.2024.3408080.

- **Alireza Ahmadi**, Julius Rueckin, Michael Halstead, Marija Popović, and Chris McCool. OptimWeeder: A Reinforcement Learning-based Approach to Control a Mobile Multi-Axis Weeding System. *Submitted to the Journal of Computers and Electronics in Agriculture, August 2024*

1.3 Collaborations

The following collaborations are directly connected to the contributions of this thesis and have resulted in the following peer-reviewed conference and journal publications:

- Michael Halstead, **Alireza Ahmadi**, Smitt Claus, Oliver Schmittmann, and Chris McCool. Crop agnostic monitoring driven by deep learning [52]. *Frontiers in plant science, 12, 2021.*
  DIO: 10.3389/fpls.2021.786702.

- Smitt Claus, Michael Halstead, **Alireza Ahmadi**, and Chris McCool. Explicitly incorporating spatial information into recurrent networks for agriculture [99]. **Received the best Agri-Robotics paper award at IEEEs**

2022 Intelligent Robots and Systems conference (IROS 2022). *IEEE Robotics and Automation Letters, 7(4):1001710024, 2022.*
DOI: 10.1109/LRA.2022.3188105.

- Anna Massfeller, Marie Zingsheim, **Alireza Ahmadi**, and Hugo Storm. Optimal design of payments for ecosystem services in the era of weeding robots. *ECCB, 01, 2024.*

- Anna Massfeller, Marie Zingsheim, **Alireza Ahmadi**, and Hugo Storm. Action-or results-based payments for ecosystem services in the era of smart weeding robots? [70] *Biological Conservation, 11099, 2025.*
  DOI: 10.1016/j.biocon.2025.110998.

- Stefan Paulus, Thomas Linkugel , **Alireza Ahmadi**, Arno Ruckelshausen, Chris McCool, Anne-Katrin Mahlein. A generalized concept of autonomy levels for weeding robots. *Submitted to Journal of Field Robotics, August 2024.*

- Felix Esser, Elias Ariel Marks, Federico Magistri, Jan Weyler, Simon Bultmann, Tobias Zaenker, **Alireza Ahmadi**, Michael Schreiber, Heiner kuhlmann, Christopher Steven McCool, Marija Popovic, Cyrill Stachniss, Sven Behnke, Maren Bennewitz, Lasse Klingbeil. Automated Leaf-Level Inspection of Crops in Agricultural Fields by Combining Aerial and Ground Robot Systems [41]. *IEEE International Conference on Robotics and Automation - ICRA40, 09, 2024.*

Chapter 2

Basic Techniques of Precision Weed Management in Arable Farms

This chapter is dedicated to explaining the fundamental techniques used in our proposed precision agriculture systems. These foundational techniques form the building blocks for the autonomous precision weed management and crop monitoring systems discussed in subsequent chapters. Topics covered include autonomy in aid of precision agriculture, vision-based arable farm monitoring, and robotic in-field targeted intervention.

2.1 Autonomy in Aid of Precision Agriculture

Achieving robust and reliable autonomy in agricultural environments requires the integration of several fundamental techniques. These methods are essential to ensure precise field localization, enabling safe navigation across complex and dynamic terrains, and informed, strategic decision-making tailored to specific agricultural tasks. The required algorithms include sensor fusion techniques, real-time signal and data processing, and environmental mapping to maintain accuracy and efficiency in operations. In the following, we briefly explain the methods that were used in this thesis.

2.1.1 Camera Model

Our robotic platform BonnBot-I incorporates multiple RGB-D and stereo cameras to perceive the environment's geometry, making it essential to understand the mathematical models underlying their operation. We use the pinhole projection model [105], which mathematically represents the imaging process through a projection function, describing how a 3D point in the environment is mapped onto the camera's 2D image plane. This projection is determined by the camera's intrinsic parameters, such as focal length, optical center, and lens distortion. The main elements of the model are shown in Figure 2.1.

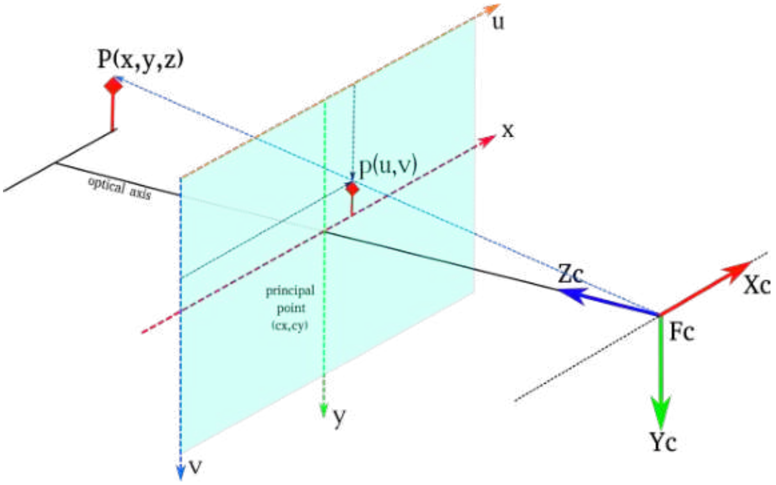


Figure 2.1: Pinhole camera model; Pinhole camera model, showing 3D point P which is projected into 2D space of image plane with pixel coordinates p.

In Figure 2.1, a camera with the center of projection denoted with O and the principal axis parallel to the Z_c axis going through the image plane is shown. The image plane is placed along the Z_c axis at a distance equal to the focal length f away from the center of the projection O. Using this model, a 3D point $\mathbf{P}(x, y, z)$ will project into the image plane forming point $\mathbf{p}(u, v)$, according to equation below:

$$\mathbf{p}(u, v) = \pi(K\mathbf{P}) \tag{2.1}$$

where $\pi(w)$ represented the homogeneous normalization and the calibration matrix $\mathbf{K}$ defines as follows:

$$\begin{aligned} \mathbf{K} &= \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \\ \pi(v) &= \frac{1}{v_z} \begin{pmatrix} v_x \\ v_y \\ 1 \end{pmatrix} \end{aligned} \tag{2.2}$$

with f_x, f_y as focal lengths in x and y direction and the principal point c_x, c_y respectively in the x and y axis (in pixels).

2.1.2 Kalman Filter

The Kalman Filter (KF) is an algorithm for estimating the state of a dynamic system, such as position and velocity, from noisy measurements. It is particularly useful for sensor fusion in agricultural robotics, combining GPS, Inertial Navigation Systems (INS), and odometry data for accurate localization. This is essential for autonomous robots operating in outdoor environments with varying terrain and GPS signal quality and is a part of the real-time process on BonnBot-I. The Kalman Filter assumes a linear system with Gaussian noise and operates in two main steps:

Initially, in the "*Prediction Step*", the KF predicts the next state based on the systems motion model:

$$\widehat{\mathbf{x}}_t^- = \mathbf{F} \cdot \widehat{\mathbf{x}}_{t-1} + \mathbf{B} \cdot \mathbf{u}_t, \tag{2.3}$$

$$\mathbf{P}_t^- = \mathbf{F} \cdot \mathbf{P}_{t-1} \cdot \mathbf{F}^\top + \mathbf{Q}, \tag{2.4}$$

where, $\widehat{\mathbf{x}}_t^-$ represents the predicted state at time t, and the state transition model is denoted by $\mathbf{F}$. The control input model is represented by $\mathbf{B}$, and the control input, $\mathbf{u}_t$, includes variables such as velocity and steering. The predicted error covariance is expressed as $\mathbf{P}_t^-$, while $\mathbf{Q}$ represents the process noise covariance, which accounts for uncertainties in the system's dynamics. After the prediction spate, which relies on the system model and the previous state of the system, the KF refines the predicted system state using sensor measurements, This step is called the "*Update Step*":

$$\mathbf{K}_t = \mathbf{P}_t^- \cdot \mathbf{H}^\top \cdot (\mathbf{H} \cdot \mathbf{P}_t^- \cdot \mathbf{H}^\top + \mathbf{R})^{-1}, \tag{2.5}$$

$$\widehat{\mathbf{x}}_t = \widehat{\mathbf{x}}_t^- + \mathbf{K}_t \cdot (\mathbf{z}_t - \mathbf{H} \cdot \widehat{\mathbf{x}}_t^-), \tag{2.6}$$

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t \cdot \mathbf{H}) \cdot \mathbf{P}_t^-, \tag{2.7}$$

where $\mathbf{K}_t$ represents the Kalman gain, which is used to update the predicted state. The measurement model is denoted by $\mathbf{H}$, and $\mathbf{R}$ represents the measurement noise covariance, accounting for uncertainties in the measurements. The measurement at time t is given by $\mathbf{z}_t$, and $\mathbf{I}$ refers to the identity matrix, which is used in various matrix operations in the Kalman filter process.

2.1.3 Extended Kalman Filter (EKF)

In reality, none of the captured sensory data or system models are linear, hence, a non-linear variant of KF is required to enable precise sensory function and state estimation. The Extended Kalman Filter(EKF) extends the KF to handle non-linear systems, which are common in real-world robotics. It linearized the system around the point of operation and integrates changes using the Jacobian of the measurement model. Generally, for non-linear state transition functions:

$$\widehat{\mathbf{x}}_t^- = f(\widehat{\mathbf{x}}_{k-1}, \mathbf{u}_t), \tag{2.8}$$

$$\mathbf{P}_t^- = \mathbf{F}_t \cdot \mathbf{P}_{k-1} \cdot \mathbf{F}_t^\top + \mathbf{Q}, \tag{2.9}$$

where, f denote the non-linear state transition function and Jacobian of f is defined as $\mathbf{F}_t = \frac{\partial f}{\partial \mathbf{x}}$. Similarly, for non-linear measurement functions:

$$\mathbf{z}_t = h(\widehat{\mathbf{x}}_t^-), \tag{2.10}$$

$$\mathbf{H}_t = \frac{\partial h}{\partial \mathbf{x}}, \tag{2.11}$$

$$\mathbf{K}_t = \mathbf{P}_t^- \cdot \mathbf{H}_t^\top \cdot (\mathbf{H}_t \cdot \mathbf{P}_t^- \cdot \mathbf{H}_t^\top + \mathbf{R})^{-1}, \tag{2.12}$$

$$\widehat{\mathbf{x}}_t = \widehat{\mathbf{x}}_t^- + \mathbf{K}_t \cdot (\mathbf{z}_t - h(\widehat{\mathbf{x}}_t^-)), \tag{2.13}$$

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t \cdot \mathbf{H}_t) \cdot \mathbf{P}_t^-, \tag{2.14}$$

where, non-linear measurement function is denoted by h and $\mathbf{H}_t$ is the Jacobian of h. A robust sensor fusion system is essential for achieving precise positioning of field robots in outdoor environments with limited GPS coverage or obstructed areas, where GPS signal degradation poses a significant challenge for systems that rely solely on GPS data. In contrast, an inertial measurement unit (IMU) provides high-frequency motion updates but suffers from drift over time and is prone to minor noise caused by disturbances. Similarly, wheel odometry can effectively track simple local motions but is highly susceptible to errors from wheel slippage and uneven terrain.

To address these limitations, we utilize an EKF to integrate data from these diverse input modalities, accounting for their respective uncertainties. This approach delivers accurate and reliable estimates of the robots position, orientation, and velocity. Such precision enables efficient navigation for tasks like precision weeding and field mapping in arable environments. Further details on this topic are discussed in Chapter 3.

2.1.4 Non-Linear Least Squares Optimization

Non-linear least-squares optimization is an unconstrained optimization technique that fits a set of m observations with a model that can be expressed with n unknowns non-linearly, where $m \geq n$. The general form of this optimization method is shown in Equation (2.15).

$$\mathbf{F}(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^{m} \mathbf{f}_i(\mathbf{x})^2 = \frac{1}{2} \mathbf{f}(\mathbf{x})^\top \mathbf{f}(\mathbf{x}) \tag{2.15}$$

where $\mathbf{f}_i(x)$ is the difference between one of the desired and predicted values in the system, and the objective function $\mathbf{F(x)}$ is defined based on the sum of the differences of the squares. A typical way to minimize this function is to find the local minimum of the function and iteratively update the estimates in the direction of descent.

2.1.5 Gauss-Newton method

While there are a bunch of different methods to minimize function Equation (2.15), we use the Gauss-Newton method, which finds $x^* = \operatorname{argmin}_x \mathbf{F(x)}$ by taking the first derivative of $\mathbf{F}(\mathbf{x})$ using Taylor expansion for small steps $\mathbf{h}$ where:

$$\mathbf{f}(\mathbf{x}+\mathbf{h}) \cong \ell(\mathbf{h}) \equiv \mathbf{f}(\mathbf{x}) + \mathbf{J}(\mathbf{x})\mathbf{h} \tag{2.16}$$

This method approximates $\mathbf{f}$ by linearizing it around x and assumes the function $\mathbf{f}$ to be locally quadratic. By inserting Equation (2.16) into Equation (2.15) we get:

$$\begin{aligned} \mathbf{F}(\mathbf{x}+\mathbf{h}) \cong L(\mathbf{h}) &\equiv \frac{1}{2}\ell(\mathbf{h})^\top \ell(\mathbf{h}) \\ &= \frac{1}{2}\mathbf{f}^\top\mathbf{f} + \mathbf{h}^\top\mathbf{J}^\top\mathbf{f} + \frac{1}{2}\mathbf{h}^\top\mathbf{J}^\top\mathbf{J}\mathbf{h} \\ &= F(\mathbf{x}) + \mathbf{h}^\top\mathbf{J}^\top\mathbf{f} + \frac{1}{2}\mathbf{h}^\top\mathbf{J}^\top\mathbf{J}\mathbf{h} \end{aligned} \tag{2.17}$$

Given the Jacobian matrices $\mathbf{J}$, we can now solve for minimizing the error, which this procedure iteratively reduces the sum of squared errors toward the minimum of the quadratic function $\mathbf{f}$ with steps of the size of $\mathbf{h}$.

$$(\mathbf{J}^\top\mathbf{J})\mathbf{h} = -\mathbf{J}^\top\mathbf{r} \tag{2.18}$$

where $\mathbf{r}$ is known as residuals and shows the actual error value at the given point $\mathbf{x}$. To update the value of $\mathbf{x}$ Equation (2.19):

$$\mathbf{x}_{t+1} = \mathbf{x}_{t-1} + \mathbf{h} \tag{2.19}$$

2.1.6 Visual-Servoing Based Robot Guidance

Visual servoing is a widely recognized technique for enhancing the accuracy and flexibility of vision-based robotic systems. This approach integrates vision-derived features directly into the feedback loop of a robotic control system, enabling reactive behavior through rapid image processing. It utilizes one or more cameras to perceive the visual information from the surrounding environment continuously, guiding the robot's movements toward a specified target [34]. In this thesis, we used visual serving to guide our precision farming robot through lanes of crop in an outdoor environment, relying only on camera observation, explained in Chapter 4. In visual servoing, the robotic task is defined by image features extracted from the target object, which are then used to control the robot's or camera's motion within the environment. The goal of the positioning task is to minimize the error function, which is designed to regulate the robot's end-effector or orientation position relative to either a global coordinate system or the reference frame of the object being manipulated. Based on visual features extracted in real-time, the error function is crucial for determining the robot's configuration and adjusting it over time to achieve precise control. Various methods have been developed to control robots using visual information, with two primary approaches emerging: position-based visual servoing (PBVS) and image-based visual servoing (IBVS).

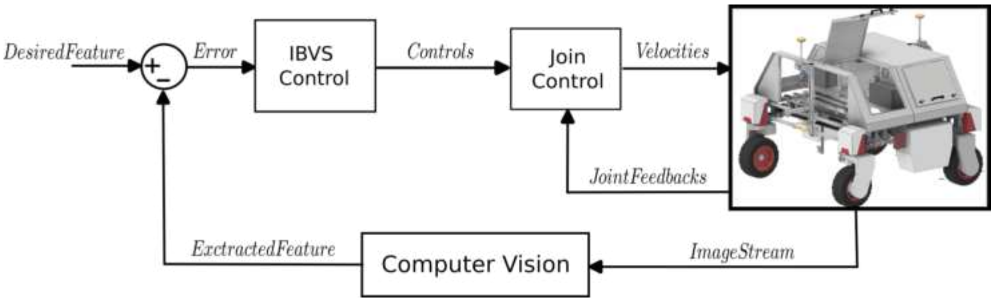


Figure 2.2: Visual-Servoing general scheme; A set of *Extracted Features* from the robot's camera, which observes the local environment, is compared to the *Desired Features* as reference. The IBVS controller then generates joint control signals in a closed-loop system to minimize the error between the extracted and desired features.

In PBVS, image data is processed to estimate the 3D pose between the camera and the target. In contrast, the IBVS calculates the error directly from features observed in the image. The error function in IBVS is defined as the difference between the desired and current image features. For a set of n image features $\mathbf{s}$, the error is:

$$e = \mathbf{s}_{\text{desired}} - \mathbf{s}_{\text{current}} \tag{2.20}$$

Where $\mathbf{s}_{\text{desired}}$ and $\mathbf{s}_{\text{current}}$ are the vectors of desired and current image features, respectively. In IBVS, the control law is designed to minimize the error in the image space. The robot velocity is related to the error and the Jacobian matrix J_s, which maps image feature velocities to robot joint velocities. The control law is given by:

$$\dot{X} = -\lambda J_s^{-1} e \tag{2.21}$$

Where $\dot{X}$ is the robot velocity in joint space, J_s^{-1} is the inverse of the image Jacobian, which relates the motion of image features to the robot's joint space velocity, and λ is a scaling coefficient. The Jacobian matrix J_s in IBVS relates the velocity of the image features $\mathbf{s}$ to the robots joint velocities:

$$\dot{\mathbf{s}} = J_s \dot{X} \tag{2.22}$$

Where $\dot{\mathbf{s}}$ is the velocity of the image features, and $\dot{X}$ is the joint space velocity. Hence, two key factors significantly influence the performance of any visual servoing method: the choice of visual features used as input for the control law and the design of the control scheme itself. The selected set of features can result in different behaviors, this variability means that selecting specific features or control schemes can sometimes lead to stability and convergence issues.

2.2 Vision-Based Arable Farm Monitoring

An advantage of surveying an agricultural scene over a pedestrian scene is the structured and relatively static nature of crops, particularly with respect to the moving platform. While the scene is somewhat static and structured, there are a number of challenges an automatic agent needs to overcome due to the complicated nature of the scene, including, illumination variation, and occlusion. Despite this structured nature, to further advance agricultural robotics, two equally important components need to be considered: intelligent perception and control methods, and abundant labeled data. Recent advances in agricultural robotics have exploited convolutional neural networks (CNNs) [59] to both alleviate some of these challenges and achieve high performance.

Modern machine learning techniques rely on CNNs to perceive useful visual information about a scene. Most of the successful deep learning techniques utilize a paradigm of multi-layer representation learning from which semantic segmentation [59] [60] has evolved. These segmentation networks can classify on a pixel level [65] the appearance of a specific class, creating class-based output maps. From a spatio-temporal perspective, RNNs [55] can exploit previous information to improve performance in the current frame. Despite these advances, only feed-forward networks have been predominately used to generate the network parameters in each layer [45]. In agriculture, one of the few examples that uses spatio-temporal information is [66], where they exploit the regular planting intervals of crop rows.

These advances in perception methods directly support the diverse vision-based tasks critical to precision agriculture. The techniques used in scene understanding in agriculture based on machine vision techniques could be categorized as follows:

1. **Object Detection**: Object detection identifies and localizes specific objects within an image by placing bounding boxes around them. This tech-

nique is commonly used for object counting, size estimation, or analyzing spatial relationships. See Figure 2.3a for an example. Popular models include:

- **Faster R-CNN**: A two-stage model that proposes regions of interest and then refines them for accurate classification and localization [85].
- **YOLO (You Only Look Once)**: A real-time object detection model that processes the entire image in one pass, balancing speed and accuracy [83].
- **DETR (DEtection TRansformer)**: A transformer-based model that simplifies object detection into an end-to-end task using attention mechanisms [26].

2. **Semantic Segmentation**: Semantic segmentation assigns a label to each pixel in an image, creating a mask where each region corresponds to a specific category, such as crops, weeds, or soil. This technique is ideal for precise boundary delineation and area measurement. An example is shown in Figure 2.3b. Key models include:

    - **U-Net**: Known for its encoder-decoder structure with skip connections, widely used in medical and agricultural imaging [91].
    - **DeepLab**: Uses dilated convolutions for capturing multi-scale context and CRFs for sharper boundaries [29].
    - **SegFormer**: Combines transformers and convolutional techniques for lightweight and high-performance segmentation [120].

3. **Instance Segmentation**: Instance segmentation identifies individual objects in an image by assigning unique pixel masks and bounding boxes to each (see Figure 2.3c). This is particularly useful for tracking or analyzing multiple entities in a scene. Notable models include:

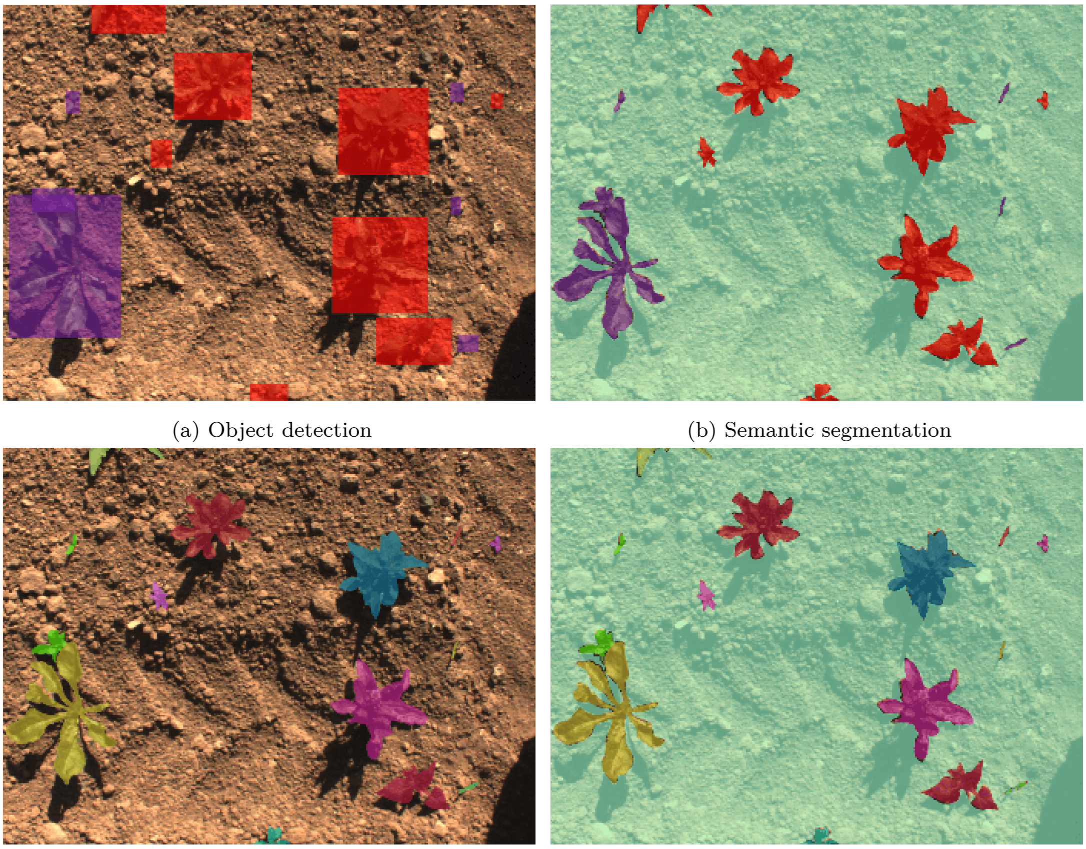
    - **Mask R-CNN**: Extends Faster R-CNN by adding a branch to generate precise object masks [54].
    - **YOLACT**: A single-stage model that predicts mask prototypes and coefficients for efficient mask generation [24].

4. **Panoptic Segmentation**: Panoptic segmentation combines semantic and instance segmentation, labeling every pixel in an image while distinguishing between "stuff" (e.g., soil) and "things" (e.g., plants), shown in Figure 2.3d. This approach provides a comprehensive understanding of agricultural fields. Key models include:

- **Panoptic DeepLab**: Builds on DeepLab to integrate both semantic and instance predictions [30].

- **Mask2Former**: A transformer-based model capable of handling semantic, instance, and panoptic tasks in a unified framework [31].

By leveraging these approaches, robotic systems can transform raw visual data into actionable insights, ensuring robust performance in tasks like crop monitoring, weed identification, and field monitoring. These techniques are essential for robotic systems in precision agriculture, enabling tasks like weed identification, crop health monitoring, and resource optimization.



(a) Object detection

(b) Semantic segmentation

(c) Instance segmentation

(d) Panoptic segmentation

Figure 2.3: Illustrative outputs from four distinct computer vision tasks applied to an image taken by BonnBot-I in CKA from a sugar beet farm. (a) shows basic object detection, where each crop is enclosed within a bounding box. (b) presents semantic segmentation, using distinct colors to represent key elements: red for weeds, purple for crops, and light cyan for soil. (c) showcases instance segmentation, focusing on crops and weeds, with each instance color-coded distinctly. (d) displays a panoptic segmentation output, categorizing crops as 'things' with sub-classes based on crop types, while all other elements are assigned to the 'stuff' category, shown in light cyan.

2.2.1 Machine Learning for on Scene Understanding

Machine learning (ML) is a field of artificial intelligence that focuses on building systems that can learn from data and improve their performance over time. Instead of following explicit instructions for every task, ML models are trained using examples and patterns in the data to make predictions or decisions. This flexibility makes ML applicable to a wide range of tasks, from detecting objects in images to predicting stock prices. Also in this thesis, we use multi-layer perceptrons (MLPs) and DNNs in different methods from environment perception to regulating actions of weeding nozzles.

A critical component of ML is the neural network, which is loosely inspired by the biological neurons in the human brain. Neural networks are powerful models designed to approximate complex functions by learning patterns and relationships in data. They are particularly useful when the data involves non-linear relationships, where traditional methods represent regression fall short.

Basics of a Neural Network

A basic neural network consists of interconnected units called *neurons* organized into layers (I) **Input Layer**: Represent the raw input features (e.g., pixel values of an image, numerical data). (II) **Hidden Layers**: Intermediate layers where computations and feature extraction occur. (VI) **Output Layer**: Produces the final output, such as a class label or a regression value.

Each neuron in the network transforms its inputs into an output using a weighted sum followed by an activation function. For a neuron j in a hidden or output layer, the output z_j is computed as:

$$z_j = \sigma\left(\sum_i w_{ij}x_i + b_j\right), \tag{2.23}$$

where, x_i are the inputs to the neuron, w_{ij} are the weights associated with the inputs, b_j is the bias term, $\sigma(\cdot)$ is the activation function. The activation function $\sigma(\cdot)$ introduces non-linearity, enabling the network to learn complex mappings. Common activation functions include:

1. **Sigmoid**:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2.24}$$

   Sigmoid maps inputs to the range $[0, 1]$, often used in binary classification tasks.

2. **ReLU (Rectified Linear Unit)**:

$$\sigma(x) = \max(0, x) \tag{2.25}$$

ReLU is widely used in hidden layers because of its simplicity and efficiency.

3. **Softmax** (used in output layers for classification):

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{2.26}$$

Softmax converts raw outputs into probabilities, ensuring the sum of outputs equals 1.

Forward Propagation

In a neural network, data flows from the input layer through the hidden layers to the output layer. This process is called *forward propagation.* For an input vector **x**, the network computes the following sequentially:

1. For the first hidden layer:

$$\mathbf{z}^{(1)} = \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}, \tag{2.27}$$

$$\mathbf{a}^{(1)} = \sigma(\mathbf{z}^{(1)}) \tag{2.28}$$

Here, $\mathbf{W}^{(1)}$ represents the weight matrix, $\mathbf{b}^{(1)}$ is the bias vector, $\mathbf{z}^{(1)}$ is the linear transformation, and $\mathbf{a}^{(1)}$ is the activation output.

2. For subsequent layers:

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}, \tag{2.29}$$

$$\mathbf{a}^{(l)} = \sigma(\mathbf{z}^{(l)}) \tag{2.30}$$

This computation repeats until the output layer.

3. For the output layer: The output depends on the type of problem. For regression, the output is typically a raw value. For classification, softmax or sigmoid activation is applied.

Loss Function

The networks performance is evaluated using a *loss function*, which measures the difference between the predicted output and the actual target. Common loss functions include:

1. **Mean Squared Error (MSE)** for regression:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{2.31}$$

where y_i is the true value and $\hat{y}_i$ is the predicted value.

2. **Cross-Entropy Loss** for classification:

$$\mathcal{L} = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} y_{ij} \log(\hat{y}_{ij}), \tag{2.32}$$

where y_{ij} is the true probability (usually 1 for the correct class, 0 otherwise), and $\hat{y}_{ij}$ is the predicted probability for class j.

Backpropagation and Weight Updates

To improve its predictions, the network adjusts its weights and biases through a process called *backpropagation.* This involves:

1. **Computing Gradients**: The derivative of the loss with respect to each weight and bias is calculated using the chain rule of calculus.

2. **Gradient Descent**: The weights are updated using an optimization algorithm. In its simplest form, gradient descent updates weights as:

$$w_{ij} \leftarrow w_{ij} - \eta \frac{\partial \mathcal{L}}{\partial w_{ij}}, \tag{2.33}$$

where η is the learning rate, and $\frac{\partial \mathcal{L}}{\partial w_{ij}}$ is the gradient of the loss with respect to the weight.

The MLPs are the simplest type of feedforward neural networks, yet they remain foundational in machine learning. Their ability to model non-linear relationships, coupled with their straightforward implementation, makes them a valuable tool for solving structured data problems and an excellent starting point for understanding deeper and more advanced architectures.

Convolutional Neural Networks(CNNs)

Modern machine learning techniques rely on CNNs to perceive useful visual information about a scene. Most of the successful deep learning techniques utilize a paradigm of multi-layer representation learning from which semantic segmentation has evolved. In a convolutional layer, the input is convolved with a set of learnable filters (kernels) to extract features. For a 2D input I and a filter K, the convolution operation is given by:

$$S(i,j) = \sum_{m} \sum_{n} I(i+m, j+n) K(m,n), \tag{2.34}$$

where, $I(i+m, j+n)$ is the input patch, $K(m,n)$ is the kernel, $S(i,j)$ is the resulting feature map. The convolution operation is followed by applying a non-linear activation function like ReLU:

$$S'(i,j) = \max(0, S(i,j)). \tag{2.35}$$

Padding ensures the output size remains controlled, particularly when edge information is critical. If p is the padding size, the input is extended before applying the convolution. **Stride** determines the step size of the filter as it moves over the input. If stride is s, the filter skips $s-1$ elements in each step. The output size of a convolution layer is computed as:

$$\text{Output Size} = \frac{\text{Input Size} - \text{Kernel Size} + 2p}{s} + 1. \tag{2.36}$$

Pooling reduces the spatial dimensions of the feature maps, retaining essential information and making the computation more efficient. The most common pooling methods are:

- **Max Pooling**:
$$P(i,j) = \max_{m,n} S(i+m, j+n), \tag{2.37}$$
where $S(i+m, j+n)$ is a patch of the feature map.

- **Average Pooling**:
$$P(i,j) = \frac{1}{k^2} \sum_m \sum_n S(i+m, j+n), \tag{2.38}$$
where k is the pooling window size.

2.2.2 Evaluation Metics

To evaluate our proposed methods in this thesis, we use several well-known metrics. For the task of object detection, we employ the F_1 metric, which summarizes the precision-recall curve into a single value. For semantic segmentation, we use the intersection over union (IoU), and for classification, we use confusion matrices. Each of these is briefly outlined below.

The precision-recall curve describes the performance of a two-class classifier (e.g., object detector) and can be summarized by the F_1 score. The precision $P = \frac{T_P}{T_P+F_P}$ and recall $R = \frac{T_P}{T_P+F_N}$ are defined by T_P which is the number of true positives (correct detections), F_P which is the number of false positives (false detections), and F_N which is the number of false negatives (miss detections). The value for P and R will vary as the threshold for the classifier varies, and to summarize the resultant curve, we calculate the F_1 score. This score is the point at which the precision equals the recall.

$$F_1 = 2 \times \frac{\mathrm{P} \cdot \mathrm{R}}{\mathrm{P} + \mathrm{R}}. \tag{2.39}$$

The IoU metric describes how well a semantic segmentation operates. Given the output of a system O and the ground truth GT the IoU is given by

$$\mathrm{IoU}\,(O, GT) = \frac{O \cap GT}{O \cup GT}. \tag{2.40}$$

The maximum IoU is 1.0, which indicates perfect semantic segmentation. For subclass performance, we also calculate the average accuracy based on the confusion matrix, such that,

$$\mathrm{conf}_{acc} = \frac{1}{I} \sum_{i}^{I} C_{ii}, \tag{2.41}$$

where C is an $I \times I$ confusion matrix and the accuracy is calculate by summing the diagonals and dividing by the number of rows. This provides the average accuracy of the confusion matrix where a value closer to 1.0 indicates a higher performance. Finally, for our tracking analysis, we utilize two metrics, the coefficient of determination (R^2) and the mean normalized absolute error (μNAE),

$$\mu NAE = \frac{1}{I} \sum_{i}^{I} \frac{|GT_i - P_i|}{GT_i} \tag{2.42}$$

where I is the number of rows being evaluated, GT is the ground truth, and P is the predicted count. These results are calculated on the total number of objects counted against the ground truth to allow direct comparison to [100]. It should be noted that this metric has a lower bound of zero (our desired outcome), but it is unbounded in the opposite direction. This is due to the prediction being scaled by the ground truth; if the prediction is considerably higher than the ground truth, this value can exceed 1.

2.3 Robotic In-Field Targeted Intervention

Efficient path planning is essential for real-time robotic interventions in agricultural fields, where precision and speed are critical for optimal performance. In this work, we developed a multi-headed weeding nozzle that requires a tailored path-planning algorithm to ensure accurate and timely targeted interventions. To address this, in Chapter 5, we leveraged the Traveling Salesman Problem (TSP) and its variant, the n-Cities Open Loop TSP (nOTSP), as robust frameworks for optimizing plant-specific routes [71, 69]. Additionally, in Chapter 6, Reinforcement Learning (RL) was integrated to enhance flexibility, enabling adaptive decision-making in dynamic scenarios and overcoming limitations of hand-made techniques. The following sections delve deeper into these methods.

2.3.1 Traveling Salesman Planning

The Traveling Salesman Problem (TSP) is a well-known NP-hard problem that has been widely studied in the combinatorial optimization field in computer science and operations research[71, 69]. The TSP is widely applied in real-world problems, including logistics, robotics, and precision agriculture. In this thesis, a variant of TSP is used for path planning where a weeding nozzle must visit multiple locations (plants) while minimizing travel distance or energy consumption in a time-constrained scenario. The objective of TSP is to determine the shortest possible route that allows a salesman to visit a given set of cities exactly once and return to the starting point. Formally, the TSP is defined as:

- **Input:** A set of n cities $C = \{c_1, c_2, \ldots, c_n\}$ and a distance matrix $D = [d_{ij}]$, where d_{ij} is the distance or cost between cities c_i and c_j.
- **Objective:** Find a permutation π of the cities such that the total travel distance is minimized:

$$\text{Minimize: } \sum_{i=1}^{n} d_{\pi(i),\pi(i+1)}, \tag{2.43}$$

where $\pi(n+1) = \pi(1)$ to ensure the route is a cycle.

The TSP can be formulated as an integer linear program (ILP) using binary decision variables x_{ij}:

$$x_{ij} = \begin{cases} 1 & \text{if the route includes an edge from city } i \text{ to city } j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.44}$$

The objective function minimizes the total distance of the route:

$$\text{Minimize: } \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij}. \tag{2.45}$$

Constraints Each city is visited exactly once

$$\sum_{j=1, j\neq i}^{n} x_{ij} = 1 \quad \forall i \in \{1, \ldots, n\}, \ \textit{and} \sum_{i=1, i\neq j}^{n} x_{ij} = 1 \quad \forall j \in \{1, \ldots, n\}. \tag{2.46}$$

Eliminate sub-tours using sub-tour elimination constraints:

$$u_i - u_j + n x_{ij} \leq n - 1 \quad \forall i, j \in \{2, \ldots, n\}, i \neq j, \tag{2.47}$$

where u_i is a helper variable representing the order in which city i is visited.

Since solving TSP exactly is NP-hard, various heuristics and approximation algorithms are used to find near-optimal solutions efficiently. Starting from an arbitrary city, the nearest unvisited city is selected at each step:

$$\text{Route: } c_1 \rightarrow \arg \min_{j \notin \text{Visited}} d_{ij}. \tag{2.48}$$

The origin of the TSP was devoted to a completely closed Hamilton path, which means a path that visits every node in the graph exactly once and returns to the starting point. However, in many scenarios, only n out of m cities need to be visited, leading to the *n-Cities Open Loop Traveling Salesman Problem* (nOTSP), which addresses such practical route planning needs [33]. We use this approach to plan trajectories for the independently controllable weeding nozzle of BonnBot-I (explained in Chapter 3) In this scenario, similar to nOTSP, the agent only needs to visit n cities/weeds. However, in our problem setting, we aim to maximize the number of visited nodes while considering other important criteria like cost and success rate. We use our constrained unidirectional node-graph representation as a base for solving nOTSP using dynamic programming.

2.3.2 Node Graph

A node graph is a visual representation of a set of objects (nodes or vertices) and their relationships (edges or connections). Each node represents an entity, and the edges represent relationships between them. Node graphs are widely used in computer science, network analysis, and AI for modeling structures like social networks, decision trees, or data flow. Let graph $G = (V, E)$ be a mathematically represented by V and E where:

- V: The set of nodes (vertices), e.g., $V = \{v_1, v_2, v_3\}$.
- E: The set of edges (connections) between nodes, e.g., $E = \{(v_1, v_2), (v_2, v_3)\}$.

For an undirected graph with no parallel edges, the total number of possible edges is:

$$|E| \leq \binom{|V|}{2} = \frac{|V|(|V|-1)}{2} \tag{2.49}$$

2.3.3 Weighted Unidirectional Graph

In this thesis, we use weighted undirected node graphs to model the relation between detected plants and enable proper planning between the nodes for conducting interventions (details in Chapter 5. A weighted unidirectional graph is a type of graph in which every edge is weighted based on specific measured or estimated characteristics representing cost, distance, or any other value, hence $G = (V, E, w)$, edges have a specific direction, meaning $(u, v) \neq (v, u)$, and $w : E \rightarrow \mathbb{R}$ is a weight function assigning a real number to each edge.

Adjacency Matrix Representation

The adjacency matrix A of a graph $G = (V, E)$ is a square matrix of size $n \times n$, where $n = |V|$, and each entry a_{ij} represents the presence and weight of an edge from vertex i to vertex j. The entries are defined as follows:

$$a_{ij} = \begin{cases} w_{ij}, & \text{if there is an edge from vertex } i \text{ to vertex } j, \\ 0, & \text{otherwise.} \end{cases} \tag{2.50}$$

2.3.4 Reinforcement Learning based Robotics Intervention

Reinforcement Learning (RL) is a branch of machine learning where an agent learns how to act within an environment to achieve a specific goal. Unlike supervised learning, which relies on labeled data, RL operates through trial-and-error interactions with the environment. The agent explores different actions, receiving feedback in the form of rewards or penalties, which guide it toward an optimal strategy. This iterative process enables the agent to discover the best actions to maximize cumulative rewards over time. RL can efficiently adapt and scale through the reward function design, bypassing the need for complex heuristics, as seen in classical approaches. Hence in this thesis, to achieve a more flexible path planning to control the weeding nozzle of BonnBot-I weeding tool, we viewed the problem as an RL problem that integrates the real kinematics of BonnBot-I too. Our method is elaborated in detail in Chapter 6.

In RL, the agent must balance *"exploration"*, trying new actions to gather information, and *"exploitation"*, leveraging known strategies to maximize rewards (see Figure 2.4). Establishing this balance is crucial to achieving optimal performance. RL methods have been successfully applied in diverse domains, including robotics, game-playing, recommendation systems, and industrial automation. Essentially, it can be interpreted as an expert human trained to excel in a specific task and respond effectively to various system scenarios, aiming to maximize the system's output, as modeled in [75].

2.3.5 Markov Decision Processes (MDPs)

The mathematical foundation of RL is built on *Markov Decision Processes (MDPs)*, which model sequential decision-making problems where outcomes depend partly on chance and partly on the agents actions.

Thus, we could define a Markov decision process (MDP) as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P})$, where $\mathcal{S}$ represents the set of possible states, $\mathcal{A}$ denotes the action space, $\mathcal{R}$ is the function that assigns a reward to each state, and $\mathcal{P}$ characterizes the probabilities

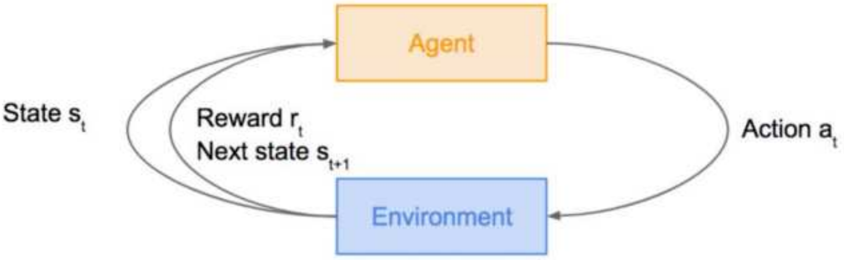


Figure 2.4: A general RL framework: the agent interacts with the environment, performing an action a_t, which leads to a new state s_{t+1} and a reward r_t that informs the learning of its policy.

of transitioning between states. The policy of an agent is symbolized by $\pi(a|s)$, indicating the likelihood of choosing action $a \in \mathcal{A}$ when in state $s \in \mathcal{S}$. The objective in an MDP context, pertinent to reinforcement learning (RL), is to identify the optimal policy that maximizes cumulative rewards over time through the cycle of action selection, state observation, and reward reception. In this thesis, we focus on an epoch-based approach, where the aggregate reward is calculated as

$$R_t = \mathcal{R}(s_t), \tag{2.51}$$

with t as the time step of epoch and s_t being the initial state drawn from a predefined distribution $\mathcal{P}_0$.

2.3.6 Deep Q-Learning (DQN)

Deep Q-Learning (DQN) is a landmark RL algorithm that enables the agent to handle high-dimensional state spaces. Traditional Q-learning, a foundational RL method, learns a value function $Q(s, a)$ that estimates the maximum cumulative reward achievable from a given state-action pair(see Figure e value function of Q-learning gets updates according to:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right], \tag{2.52}$$

where α is the learning rate, r denotes to the immediate reward of every epoch, and γ is the discount factor that regulates between exploration and learning.

DQN incorporates two critical techniques, first **Experience replay** that stores past experiences in a buffer and samples them randomly to break correlations between consecutive updates. Second, the **Target network**, A separate network that is updated less frequently to stabilize training. DQN is particularly effective in environments with discrete action spaces, such as video games and grid-world tasks, where it famously achieves human-level performance on Atari games.

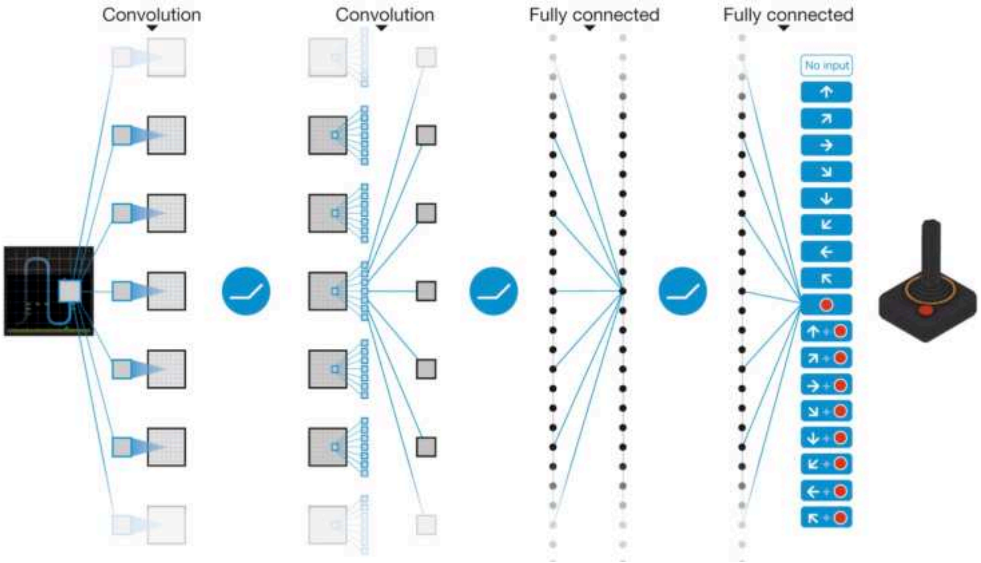


Figure 2.5: Human-level control [75]; visualization f a Deep Q Network(DQN) including (left) observation input,(middle) hidden layers, and (right) fully-connected layers and output signal.

However, Q-learning becomes impractical in environments with very large or continuous state spaces, as maintaining a table of $Q(s, a)$ values is infeasible. DQN overcomes this limitation by approximating $Q(s, a)$ using a deep neural network parameterized by θ. To train its network, the loss is defined as:

$$L(\theta) = \mathbb{E}_{(s,a,r,s')\sim\text{Replay Buffer}} \left[\left(r + \gamma \max_{a'} Q_{\theta'}(s', a') - Q_\theta(s, a) \right)^2 \right], \tag{2.53}$$

where θ' represents the parameters of a *target network* that provides stable Q-value targets.

2.3.7 Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO) is a policy-gradient method that directly optimizes the policy, rather than the value function, to maximize the expected cumulative reward:

$$\mathcal{J}(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta} \left[\sum_t \gamma^t r_t \right]. \tag{2.54}$$

PPO improves upon earlier policy-gradient methods by introducing a *clipped objective function*, which ensures stable updates by limiting the deviation of the new policy from the old policy. The clipped objective is defined as:

$$L^{\text{CLIP}}(\theta) = \mathbb{E} \left[\min \left(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right) \right], \tag{2.55}$$

where $r_t(\theta)$ is the ratio of probabilities under the new and old policies, and $\hat{A}_t$ is the advantage estimate.

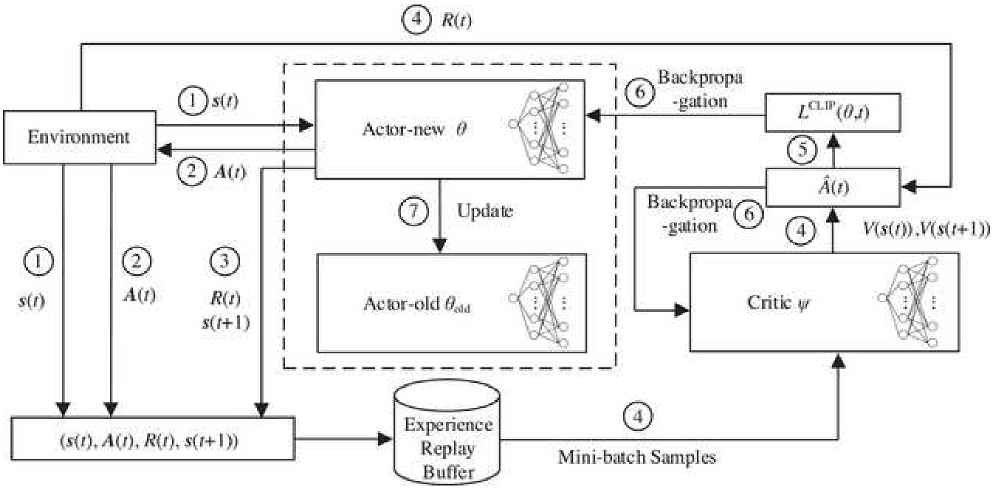


Figure 2.6: General scheme of the Reinforcement Learning framework using Proximal Policy Optimization (PPO), illustrating the agent-environment interaction loop, policy updates, and reward-driven learning process.[106]

PPO's ability to handle continuous action spaces, combined with its robustness to hyperparameter tuning, makes it highly effective for complex, dynamic environments. In this thesis, we utilize PPO to optimize the behavior of our system in a real-world agricultural scenario. The algorithm's stability and adaptability make it an ideal choice for tasks requiring precise and adaptive decision-making. Reinforcement Learning has evolved from foundational methods like Q-learning to advanced techniques like PPO, enabling it to tackle increasingly complex problems. PPO, in particular, represents a milestone in RL, balancing exploration and exploitation while ensuring stable training. Its application in this thesis demonstrates its effectiveness in optimizing systems within challenging real-world environments, contributing to the broader field of RL and its practical impacts.

Chapter 3

BonnBot-I: A Precision Weeding and Crop Monitoring System

In this chapter, we introduce BonnBot-I a platform capable of autonomous field navigation, monitoring, and weed management for arable farms. The BonnBot-I is equipped with multi-modal imagery, inertial and geometric sensors, as well as a novel weeding tool that enables plant-level interventions in arable farms, which is introduced in [5]. As Chapter 1 discussed, agricultural robotics technology must align with the practical needs of farmers, which includes reducing costs and labor associated with deploying innovative technologies. Furthermore, ensuring the effectiveness and affordability of novel farm management systems is imperative to meet the requirements of farmers, serving as both cost-

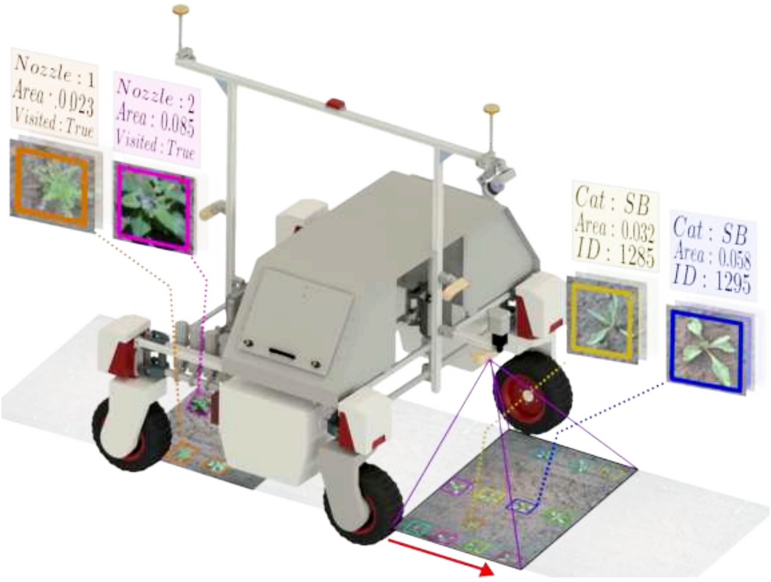


Figure 3.1: BonnBot-I Platform, a robotic platform capable of conducting field monitoring and precision weed management in an arable field.

effective platforms and replacements for manual or mechanical labor [66].

In a general sense, the capabilities of robotic systems in arable farms could be categorized into three major subjects: autonomous navigation, farm monitoring, and targeted interventions, including seeding, fertilization, weeding, etc. Hence, in this chapter, we delve into addressing the first question presented in Section 1.1, as "*What design features are essential for creating a robust, farming robot capable of crop monitoring, plant-level intervention, and autonomous navigation?*".

The advancement of precision farming, characterized by assigning specific shapes to fields, particularly multiple parallel rows, has significantly contributed to improving the autonomy of agricultural practices. It results in facilitated crop row navigation and a more constrained variety in environmental dynamics.

Nevertheless, flexibility is crucial in agricultural technologies due to the diversity of farming environments. Solutions must adapt to various fields and environmental conditions with minimal supervision and re-engineering. In recent years, smart farming has seen advancements in selective crop and weed treatments. This progress is partly due to farming sites adapting to support multiple parallel rows. This design has made cultivation, monitoring, and weeding more efficient for crops like lemon balm, mint, potatoes, and beans worldwide.

Despite these advancements, many systems lack real-time field monitoring. This capability is essential for providing detailed information about the state of crops and weeds. Without it, precision farming approaches face limitations, especially in autonomous operations that require constant environmental awareness.

With BonnBot-I, we aimed to address this gap. We introduced a novel weeding tool combined with an advanced monitoring system. The system integrates a multi-modal sensor array, using RGB-D cameras, GPS, and LiDAR. This enables precise navigation and thorough field analysis. The design improves real-time weed detection and intervention. It also enhances decision-making by providing detailed insights into crop growth and weed spread. These features make BonnBot-I capable of performing autonomous, biodiversity-aware weeding effectively. In this chapter, we make the following contributions to make:

- We **introduce BonnBot-I a fully autonomous precision weeding platform** fully compatible with ROS.
- Propose **a new concept for weeding tools** that enables flexible, high-precision weed management.
- Improve crop monitoring performance of BonnBot-I by exploiting the extra robotic vision-based and sensor fusion techniques.
- Release three new datasets collected in row-crop arable farming sites, including sugar-beet and corn crop (**SB20**, **CN20**, **SB21**).

In this chapter, we introduce BonnBot-I, a precision weeding and crop monitoring robot developed to tackle all research questions presented in Chapter 1. This novel system is equipped with advanced sensors, a powerful computation unit, and a unique intervention tool, enabling real-time, precise surveillance and intervention in farming fields. Section 3.1 outlines the key components and design philosophy of the platform. We then present simulation tools in Section 3.2, tailored for precision weeding tasks. Section 3.5.6 details three specialized datasets collected over two years at the University of Bonn's Campus Klein-Altendorf, used for development of new methods and evaluations throughout this thesis. Lastly, Section 3.4 we elaborate on the monitoring system designed and deployed on BonnBot-I which enabled instance semantic segmentation and tracking of plant-level intervention on real fields. Furthermore, we discuss techniques for improving DNN-based monitoring by leveraging geometric and motion data from BonnBot-I, which are crucial for the success of the interventions detailed in subsequent chapters.

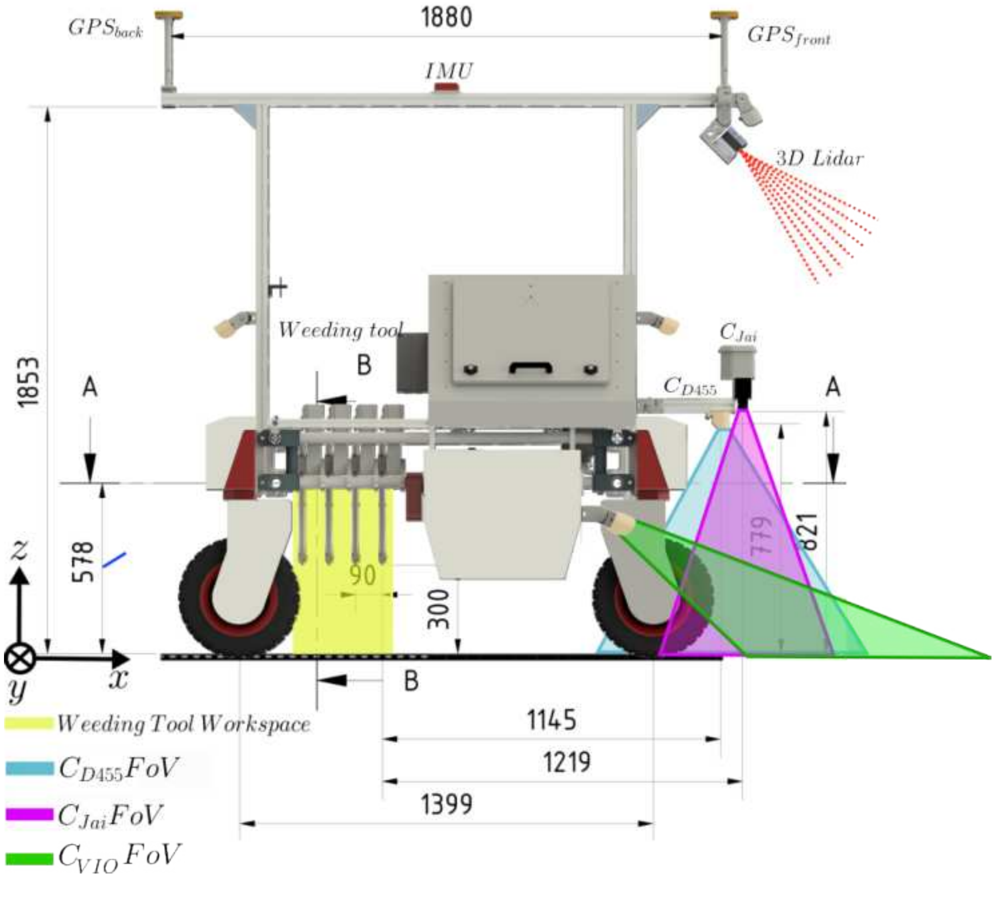


Figure 3.2: BonnBot-I dimensions and Sensory field of view setup; BonnBot-I is equipped with a range of sensors to perceive the most important information on the field in real-time. Here, the field of view of different sensors is depicted color-coded (laser scanner(red), multi-spectral Jai camera (purple), RGB-D Intel camera (light-blue), Visual odometry Camera (green), and weeding workspace in yellow.

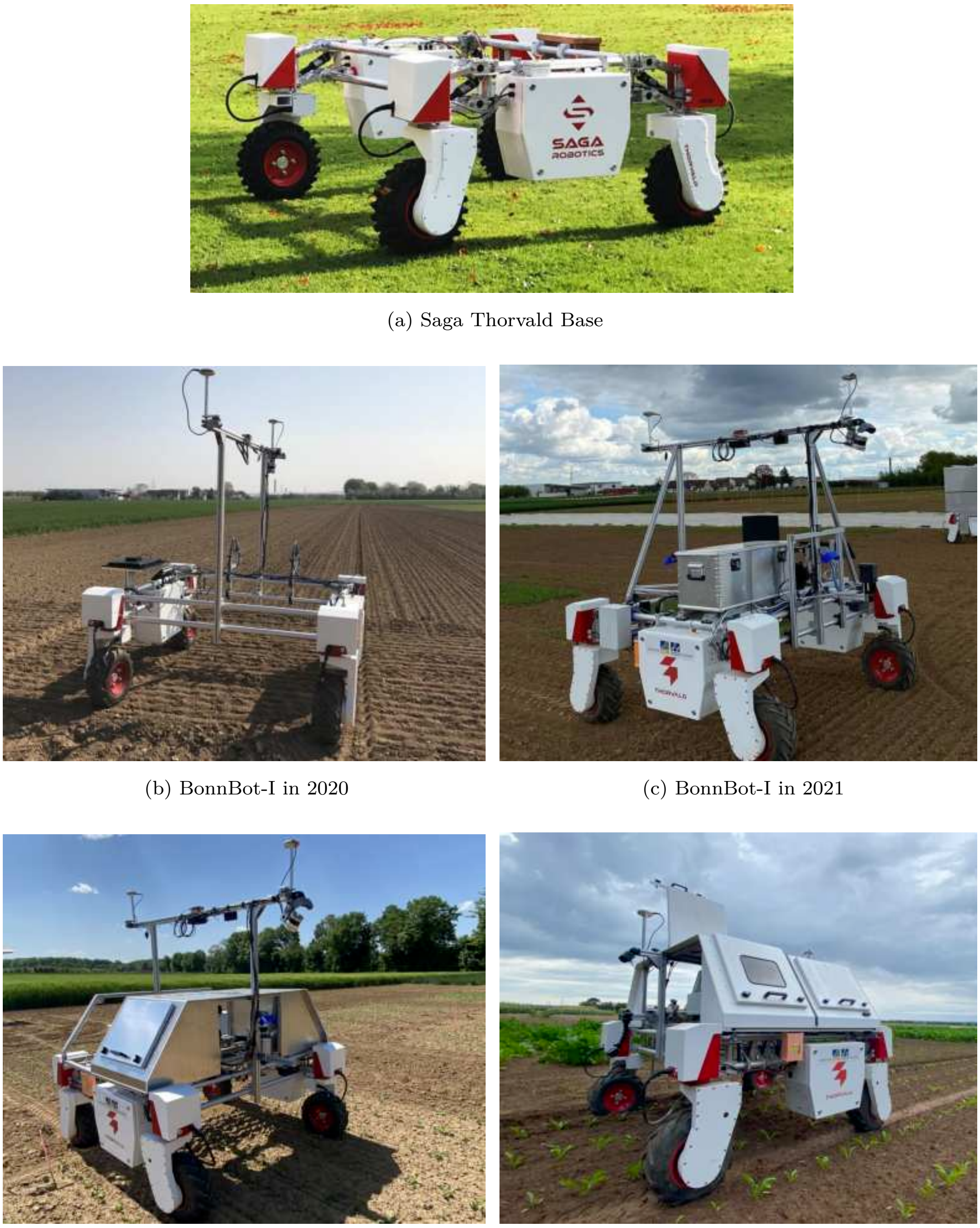
3.1 BonnBot-I A Physical System

The base platform is a Thorvald system [49], which is a lightweight four-wheel-drive (4WD) and four-wheel-steering (4WS) system. With considerable modifications, throughout four years, we adapted this platform to an arable farming and phenotyping robot suitable for operation in real-field applications like crop/weed monitoring and targeted interventions.

These modifications were carried out to ensure they met European farming standards for the distance between rows. The European phenotypic regulation pattern ($35\,cm$ or $55\,cm$ between crop rows) leads to either two or three crop rows in each lane with a total width of $1.25m$. Based on these specifications, the width from wheel-center-to-wheel-center was set to $1.5m$ (see Figure 3.2). To ensure we could monitor the different growth cycles of our primary crops (sugar beet, wheat, and corn), the vertical clearance is set to $0.57m$, and has a maximum speed of $1.5\,m/s$. The length of BonnBot-I $1.39m$, was selected to ensure there was adequate space to install replicating weeding tools. These dimensions also act to increase the stability of the platform on the uneven terrain witnessed in arable farmland. An overview of the BonnBot-I platform and four versions of BonnBot-I through the years 2020 to 2024 are depicted in Figure 3.3.

We equipped BonnBot-I with a comprehensive selection of sensors to enable autonomous navigation, crop monitoring, and precise plant-level interventions in the field Figure 3.4. These sensors are divided into two main categories: one for "localization and navigation" and another for "environment perception and intervention". This setup allows BonnBot-I to excel in arable fields, performing tasks such as precise localization (using sensor fusion and dead-reckoning techniques), waypoint-based and vision-based autonomous navigation in row-crop fields, and environment perception for selective, real-time plant-level interventions.

The sensor setup includes inertial measurement units (IMUs), a dual GNSS receiver with NRTK technology supported by Germany's SAPOS positioning system for accurate localization, as well as 2D and 3D vision sensors like NIR-RGB-D cameras. To provide a homogeneous view of the environment, we added an Ouster-OS1 lidar to the front. The OS-1 lidar is a 64-beam lidar covering 360 and 45 degree field of view in the horizontal and vertical directions, with a maximum range of $120m$. degrees horizontal and vertical field of views with a maximum range of $120m$ meters. This provides long-range sparse depth data to enable 3D mapping, navigation, and safety. We will delve into the specific components in the following sections.



(a) Saga Thorvald Base

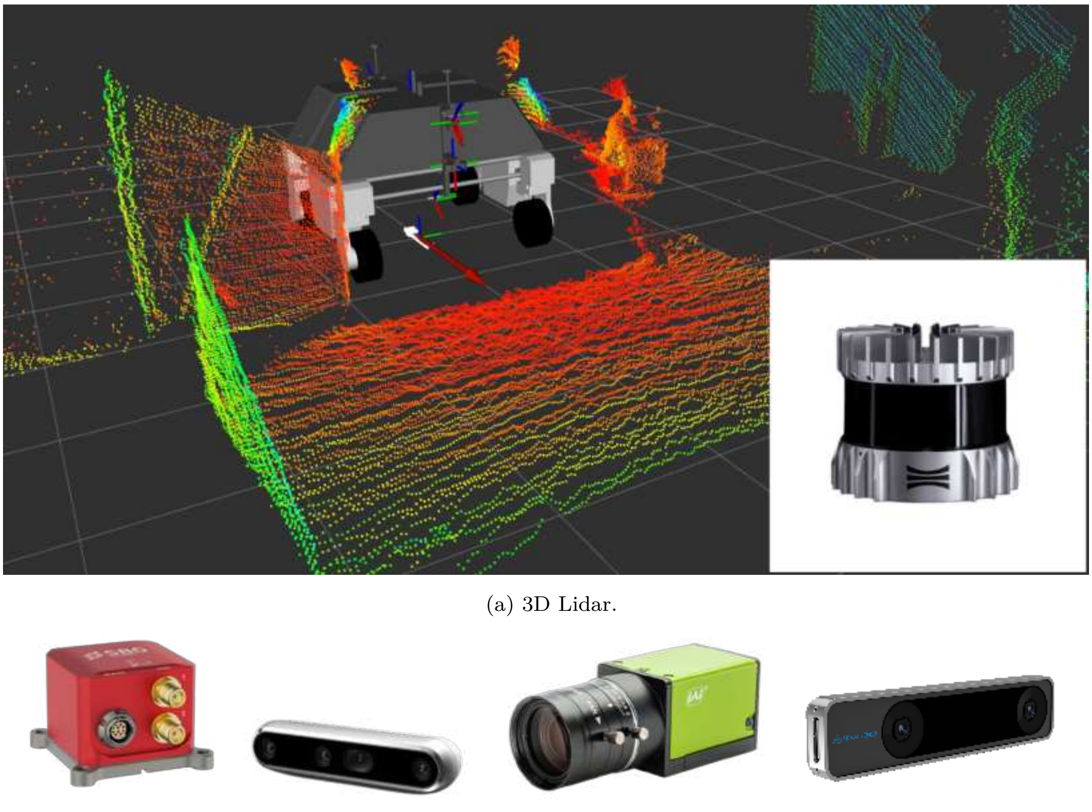
(b) BonnBot-I in 2020

(c) BonnBot-I in 2021

(d) BonnBot-I in 2022

(e) BonnBot-I in 2023-2024

Figure 3.3: BonnBot-I evolution's in five years; (a) BonnBot-Ibase; Saga thorvald an omni-directional field robotic platform [49], (b) BonnBot-I in 2020 without weeding tool and only capable of autonomous GPS based navigation and multi-modal data collection in fields, (c) BonnBot-I in 2021 equipped with one linear weeding axis equipped with a single spray nozzle and visual row-crop field navigation package, (d) In 2022, the weeding tool on BonnBot-I completed with 4 linear axes and 4 spray nozzles and it could perform crop/weed segmentation to intervene with weeds, (e) And finally in 2023-2024 BonnBot-I could reach to its full capabilities of conducting plant level bio-diversity aware weeding interventions in real fields.



(a) 3D Lidar.

(b) NRTK-GPS. (c) RGB-D-IR. (d) Multi-spectral RGB-NIR. (e) VIO Camera.

Figure 3.4: BonnBot-I sensor configuration; (a) Ouster; a 3D laser scanner, (b) Ellipse-D SBG device; an INS-GPS localization system with dual antenna using SAPOS technology providing NRTK service, (c) Real-sense D455 RGB-D-IR camera; a stereo-based camera providing on-board depth frames, (d) JAI camera; an RGB-NIR prism-based multi-spectral camera, (e) Intel T265 visual odometry camera.

3.1.1 Dead-Reckoning and Precise Outdoor Localization

Precise centimeter-level localization is a necessity to perform plant-level farm monitoring and interventions. The localization of BonnBot-I is performed with a compact inertial navigation system (INS), Ellipse2-D SBG Systems [94], which includes an IMU and dual-antenna receiver. Multi-band GNSS receivers are fixed at the front and back of the robot at a height of $1.85m$ above the ground. A high-frequency extended Kalman filter fusion Section 2.1.3 of IMU, GPS, and wheel odometry data provides us with a horizontal and vertical position accuracy of $2cm$ and $3cm$ respectively. Furthermore, the heading of the platform can be determined with an accuracy of 0.1° and 0.3° in roll-pitch and yaw directions, respectively.

3.1.2 Vision Sensors Aiding Navigation

An agricultural robotic system must deal with a range of challenges posed by the environment in the agricultural cities, including self-similar views that hinder navigation capabilities, as well as sudden variations in illumination conditions, slippery, muddy, hard, bumpy surfaces, extreme humidity, etc. Self-similarity poses a challenging issue, and at the same time, brings some benefits to automation in agriculture. This issue is due to crops being arranged in multiple parallel curves within different plots of arable fields. Hence, a camera in most configurations will see multiple rows that might not be located underneath the robot. This introduces uncertainty in identifying rows that the robot should follow.

Furthermore, other challenges persist, particularly regarding accurate crop-row navigation and precise plant localization, which are critical for successful automation. Most advanced agri-robotic systems operate in controlled agricultural environments reliant on structural information and precise real-time kinematic (RTK) GNSS receivers for autonomous navigation. Yet, the majority of fields still adhere to traditional seeding methods without auto-seeding geo-referenced systems, highlighting the need for GPS-independent navigation approaches.

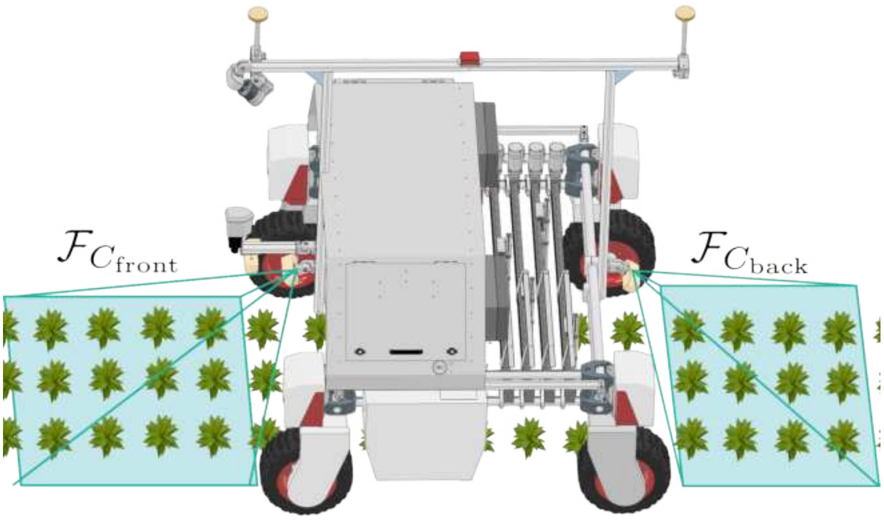

$\mathcal{F}_{C_{\text{front}}}$
$\mathcal{F}_{C_{\text{back}}}$

Figure 3.5: Navigation Cameras; BonnBot-I visual navigation sensory setup, two Intel Realsense-D435i rolling-shooter RGB-D cameras in front and back of the robot (C_f and C_b) tilted towards the crop rows underneath the robot .

In Chapter 4 we will demonstrate our approach, enabling BonnBot-I to automatically traverse a multi-crop-row field without human intervention and independent of any global localization services. To enable such an approach we added two symmetrically fixed Intel RealSense-D435i in front and back of the robot denoted by C_f and C_b in Figure 3.5. In Chapter 4 we show that this sensor configuration could reliably navigate a field regardless of the number of crop rows under the robot or the crop type; this was achieved without using any global localization service.

3.1.3 Vision Sensors Aiding Intervention

Vision sensors play a crucial role in achieving precise and efficient agricultural interventions. Advanced vision technologies, such as multi-spectral and RGB-D cameras, support tasks like precise plant phenotyping and selective weeding by ensuring minimally invasive actions in crop fields. By collecting locally observed multi-modal data on plants, their types, growth stages, and environmental factors, these sensors enable robots like BonnBot-I to identify and target specific plants for intervention with accuracy (see Figure 3.2). In the following, we briefly introduced the cameras that have been used:

1. ***JAI Camera:*** In front of the platform, an AD-130GE multi-spectral GigE camera with rolling shutter CCD is used, depicted in Figure 3.4e. It uses two CCD sensorsone for color and the other for near-infrared (NIR)

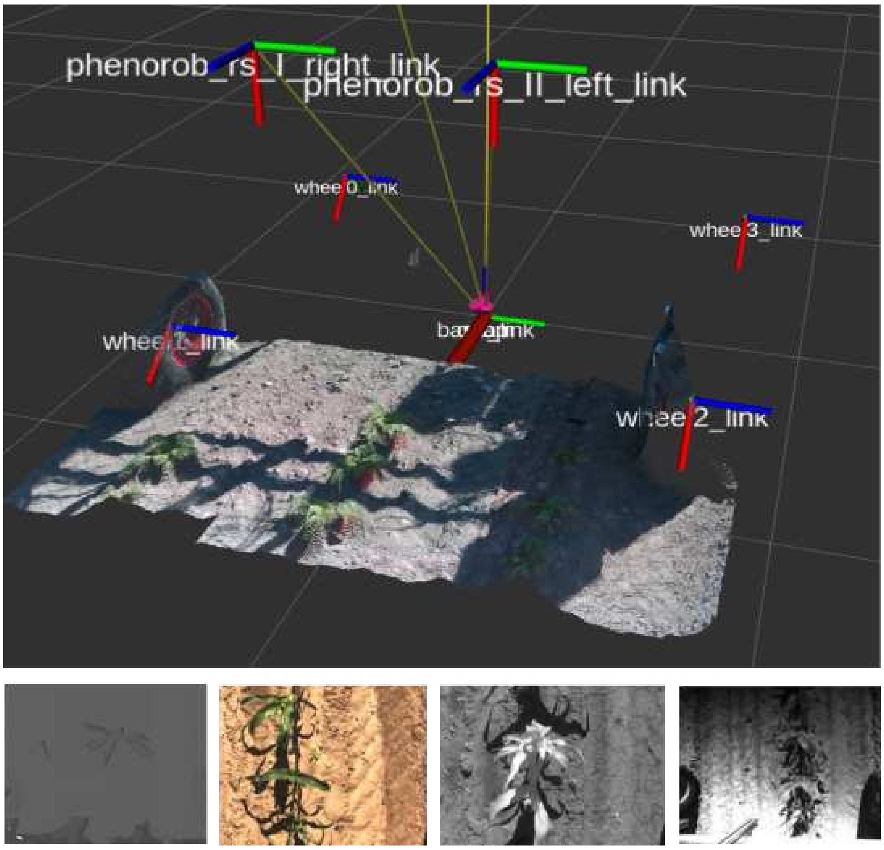


Figure 3.6: Detection Camera Sample Images (from 2020 with 2 Realsense and 2 Jai cameras in front); (top) Synchronous visualization of captured RGB and depth data from a camera of BonnBot-I in Rviz application along with TF-tree markers; (bottom left to right) depth image, RGB image, N-IR image, and normal IR image of a scene from a Corn farm in CKA.

monochrome combined with prism optics. This design allows it to capture both visible and NIR data from the same angle of view, aiding in precise vegetation health assessments and species differentiation. Figure 3.6 depicts some sample data from this sensor.

2. ***RealSense D455 Camera:*** The RealSense D455 Figure 3.4c, is a global shutter camera that provides RGB, IR, and stereo-based computed depth images. We mounted this camera with a nadir-view on BonnBot-I at a height of $0.78m$, offering a viewable area of $1.4m \times 0.81m$, which covers the space between the two front wheels as shown in Figure 3.6, resulting in a ground sampling of approximately $\sim 1mm$. This sensor is the primary tool used for field monitoring.

3. ***RealSense T265 VIO-Camera:*** To improve motion estimations of BonnBot-I we use an Intel RealSense T265 VIO camera (in green in Figure 3.6) that utilizes advanced dual fisheye lenses and a vision sensor to capture wide-angle images. It integrates features extracted from stereo vision with IMU data through its proprietary V-SLAM (Visual Simultaneous Localization and Mapping) algorithm. This approach allows the camera to map its surroundings with high precision, even in GPS-denied environments.

The data received from such devices is used to achieve instance-based semantic segmentation and provide critical information such as plant species (e.g., weed types), surface area from the RGB-D camera (translating to plant growth metrics), and the location of plant stems used for steering intervention pin-points. We developed advanced deep neural networks that process individual observations while integrating 3D geometric information, temporal data, and motion-related information measured on the platform in real-time, which are introduced in detail in Section 3.4.

3.1.4 Multi-Axes and Multi-Head Weeding Tool

Weeding is a critical task for farmers during each cultivation cycle. A major challenge in modern weeding is reducing herbicides and pesticides while maintaining crop quality and yield. To address this, developing flexible, repeatable weeding systems capable of deploying various end-effectors is a key focus for BonnBot-I. The platform's main goal is to enable precise interventions, as outlined in [5], allowing it to adapt its tools according to soil conditions and weed populations.

The proposed design features independently controlled high-resolution ZLW-1040S Igus linear actuators mounted $0.57m$ above the ground, creating a working space of $1.3m \times 0.36m$. The current setup includes four linear actuators, each

1.3m long, controlled by Igus dryve D1 motor control systems via Modbus connections. These actuators offer high precision, with a resolution of 0.01cm, and can achieve maximum speeds of 5m/s and accelerations of 10m/s^2. The system allows flexibility in tool deployment, as the linear axes, currently equipped with spot-spray nozzles, can accommodate various end-effectors, such as mechanical hoes.

A Raspberry Pi 3-B single-board computer controls linear actuators and nozzle valves, interfaced through a Pixtend v1.3 I/O board. The Pixtend board, a PLC-compliant logic controller, offers a broad range of digital and analog I/O options and standard serial interfaces like RS232, RS485, Ethernet, and CAN, shown in Figure 3.8. Each linear axis operates independently, equipped with ASCO solenoid L172V03 spray valves, featuring an On-Off time of approximately 10ms, mounted on aluminum lever arms at a height of 0.3m. High-speed N-channel MOSFET transistors (IRFZ44N 55V, 41A) control the nozzle valves, ensuring minimal delay with a total operation time of around $10 - 12ms$ per spray head.

The chemical application system includes a 8L reservoir tank for compressed liquid with up to 10bar pressure and an 8bar portable compressor fixed on the robot. Spray footprint control is achieved by adjusting the lever arm height and liquid pressure. Depending on the pressure settings, individual nozzle footprints can vary between 0.02m and 0.13m. This design provides precise and adaptable weed management capabilities, enhancing the overall efficiency of the weeding process.

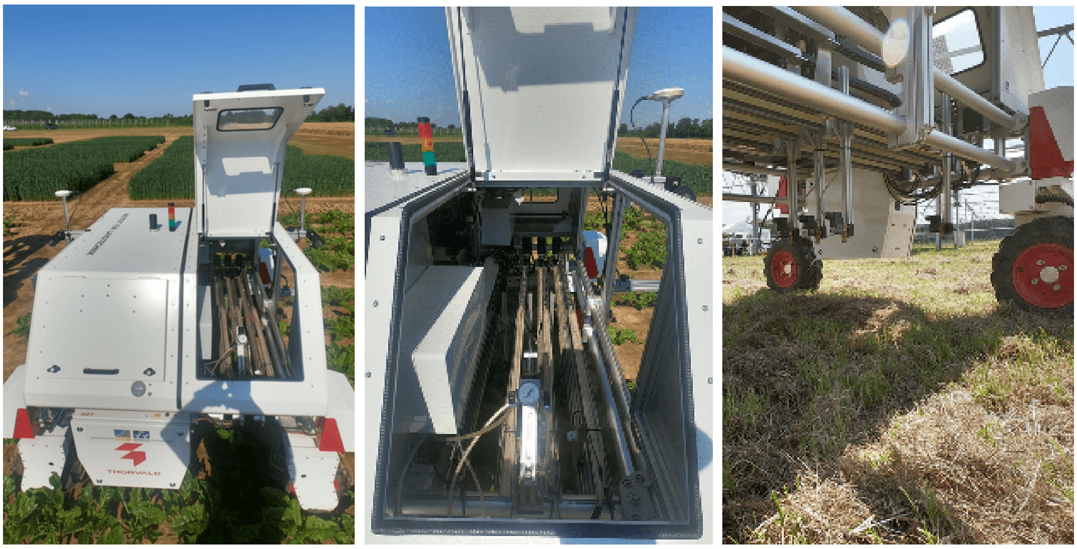


Figure 3.7: BonnBot-I Weeding Tool; the latest version of the weeding tool on BonnBot-I is equipped with four linear axes and each axis carries one spray nozzle covering the wheel-to-wheel area under the robot.

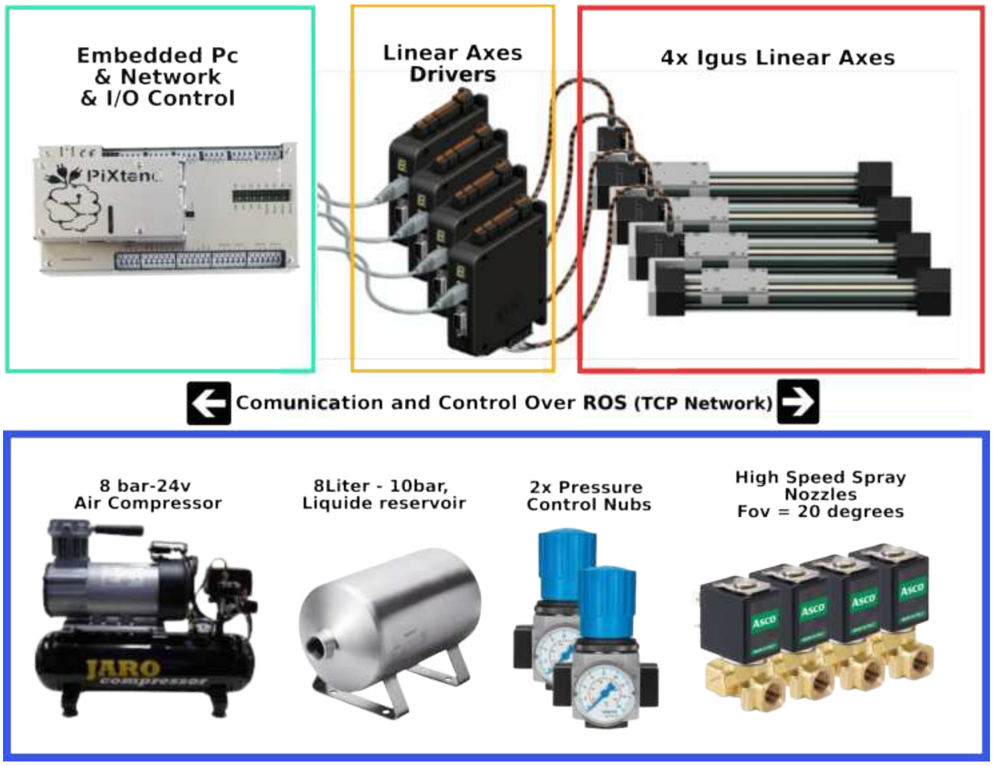


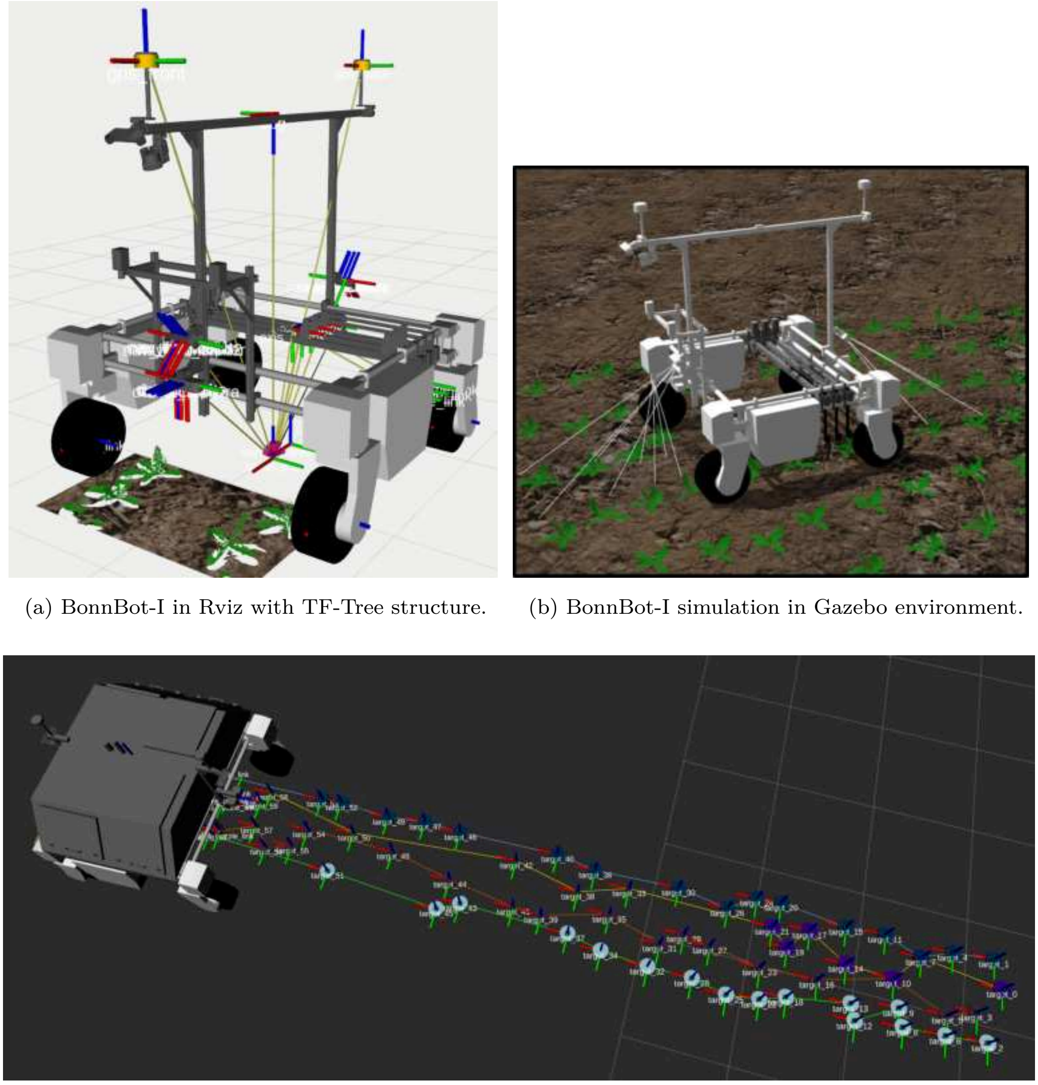
Figure 3.8: Hydraulic and Chemical application system schematic; (cyan) Pixtend Embedded Controller Board, (Orange) Igus linear axes Driver interfaced by Mud-Bus, (red) Igus Linear axes with brush-less motors and Brake system, (blue) Air compressor to pressure up piping system, herbicide reservoir supplying 8 liters of liquid for intervention, Pressure control nubs to regulate pressure on nozzles and reservoir, and High-Speed spray nozzle controlled via High-Speed MOSFETs to grantee low latency on execution of interventions.

3.2 Weeding Simulation Framework

Conducting experiments in real fields can be time-consuming and costly. To address this, we developed accurate and reliable simulation environments that closely mimic real-world conditions. These include a customized ROS-based [92] one-to-one scale simulation model for BonnBot-I and a native Python simulator specifically designed for weeding applications and intervention monitoring. The Python simulator is also compatible with OpenAI Gym [76], enabling reinforcement learning trials.

Several demonstrations of the ROS-based simulation are shown in Figure 3.9, where all sensors and actuators are fully operational. Additionally, we developed a Gazebo [47] plugin capable of generating various field configurations, including different crop sizes, weed densities, number of rows, and curvature. This allows us to simulate diverse field conditions and test navigation strategies in realistic settings Figure 3.9. The ROS-based simulation is designed to replicate the physics and dynamics of the BonnBot-I platform at its full scale. It incorporates the robots exact dimensions, sensors, and weeding actuators into a simulation package built upon the Saga [48] framework of Thorvald platform description. By matching the real-world specifications and field scenarios, simulation environments like Gazebo or Moveit allow us to test navigation strategies and weeding techniques with high reliability.

Our self-developed native Python-based simulation framework simulates the robot kinematics and can generate synthetic crop rows with varying weed distributions. We used Open3D and Pyglet python libraries for rendering graphics,



(a) BonnBot-I in Rviz with TF-Tree structure.

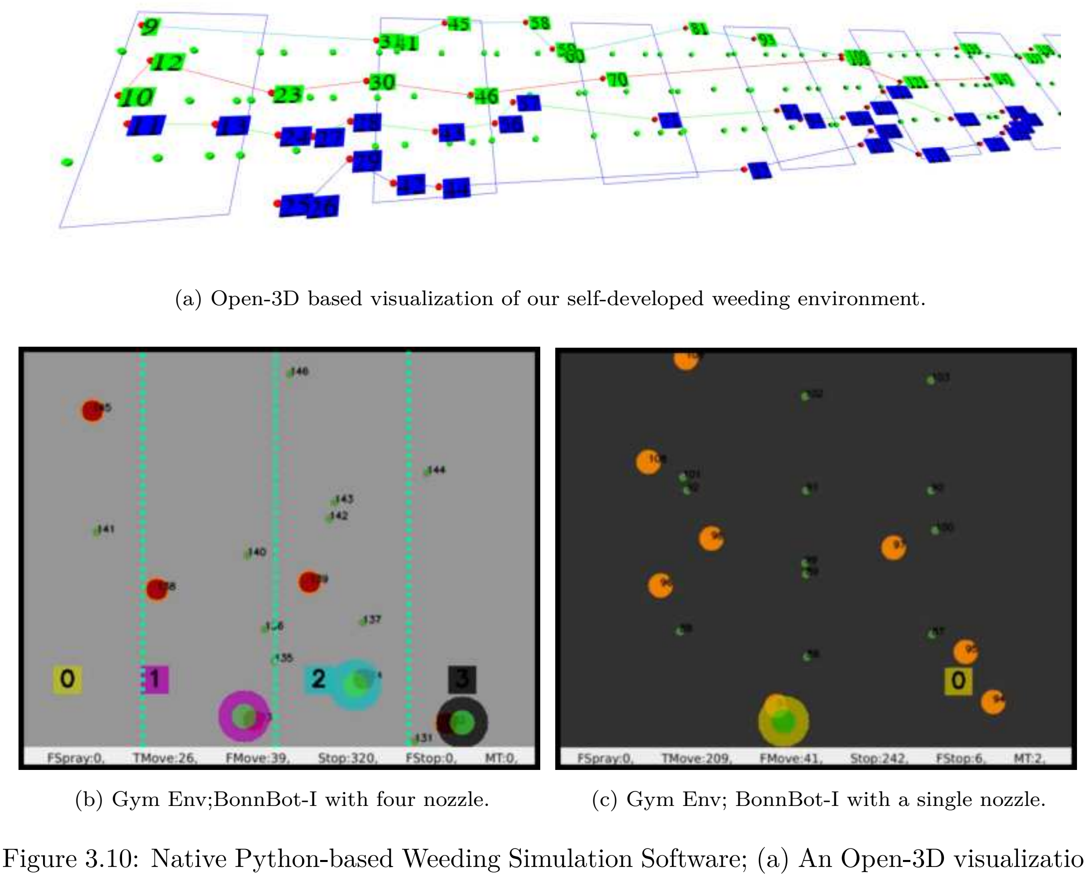
(b) BonnBot-I simulation in Gazebo environment.

(c) BonnBot-I full simulation with weeding action visualizations.

Figure 3.9: BonnBot-I ROS simulation with active sensors and actuators; (a) An example view of Rviz software of BonnBot-I conducting weeding scenario in simulation with active sensing and actuation (b) BonnBot-I simulated twin running vision-based crop row navigation, using self-developed field models in Gazebo environment, (c) A complete weeding scenario visualization in Rviz with detected and treated plant (plant centers: colored markers, circles: treatment footprints) and four execute trajectories from weeding tools of BonnBot-I.

a simplified example view of the planning scenario is shown in Figure 3.10a-(a). The field monitoring environment within the simulation framework serves as a critical component for evaluating and refining the crop and weed monitoring systems. This environment simulates a wide variety of crop conditions, plant sizes, and weed distributions, allowing us to stress-test the robot's sensing, decision-making, and weeding capabilities in highly controlled, repeatable settings.

The flexibility of the field generator enables the creation of complex crop-row structures, facilitating the development of robust algorithms for both autonomous navigation and plant-level intervention. By accurately reflecting the challenges present in real-world farming conditions, this environment allows us to fine-tune monitoring and weeding strategies before deploying them in the field. The simulations generated not only assess plant recognition and localization but also the efficacy of intervention tools under various plant densities and biodiversity considerations. The integration of an OpenAI Gym-compatible simulation framework represents a significant step forward in enhancing the robot's decision-making


0
1
2
3
FSpray:0, TMove:26, FMove:39, Stop:320, FStop:0, MT:0,
0
FSpray:0, TMove:209, FMove:41, Stop:242, FStop:6, MT:2,

(a) Open-3D based visualization of our self-developed weeding environment.

(b) Gym Env;BonnBot-I with four nozzle.

(c) Gym Env; BonnBot-I with a single nozzle.

Figure 3.10: Native Python-based Weeding Simulation Software; (a) An Open-3D visualization of a weeding operation on a real crop-row field distribution, (b,c) The image-based observation of BonnBot-I weeding gym environment; there are four agents (yellow, pink, cyan, and black squares) in a field with crops (green circles) and weeds (red circles). The spray footprint circles with a central green core indicate successful intervention.

capabilities. This environment allowed for the application of reinforcement learning (RL) techniques, enabling BonnBot-I to autonomously learn and improve its weeding strategies through trial and error, presented in Chapter 6. By utilizing a well-defined action space and reward function, the system can iteratively optimize plant-level interventions, improving its ability to plant more efficient trajectories for weeding heads of BonnBot-I weeding tool. The OpenAI Gym framework supports seamless experimentation, where various RL models can be trained and evaluated within a controlled virtual environment. This provides the robot with a platform to simulate real-world interventions while continuously improving its strategies, thereby minimizing the need for extensive field tests and allowing for more adaptive and intelligent weeding operations in complex, dynamic farming conditions, we will elaborate more on this topic in Chapter 6.

3.3 Datasets

One of the major challenges in agricultural robotics and precision farming is the scarcity of detailed, annotated datasets. To address this issue, we developed a series of specialized datasets using BonnBot-I specifically for precision agriculture applications. Over four years (2020 to 2023), in several trial sessions data was collected at the University of Bonns Klein-Altendorf campus (CKA), focusing on sugar beet, corn, and mixed oat and fava bean fields. Resulting in the creation of six datasets **SB20**, **CN20**, **SB21**, and **WeedAI** [126] published in separate publications, and two other datasets **SB22** (sugar-beet) and **Mixed-Crop 22 (MC22)** (mixed oat and fava bean), which are not published yet. Despite recording nearly 20 terabytes of data with the BonnBot-I platform, only a small portion was annotated due to the complexity and labor-intensive task. However, we successfully made the SB20, CN20, and SB21 datasets publicly available through a series of publications, the details of which are introduced below. By making

Table 3.1: Included classes in datasets; Overview of crop and weed class distributions across SB20, CN20, SB21, and SB22 datasets, including counts for different species and unknown categories.

Name	**Crop**	**Bi**	**An**	**Cy**	**Pe**	**Th**	**Ch**	**Ve**	**Po**	**Lp**	**So**	unknown
SB20	768	241	19	64	620	775	253	-	-	-	-	185
CN20	2570	8	24	14	40	859	225	8	21	-	-	28
SB21	942	4	1829	7	143	793	1135	244	51	449	43	287
SB22	200	-	-	-	-	-	15	21	-	-	38	13
MC22	8796	-	60	283	1	-	-	-	7	33	43	626

them publicly available, these data mark a crucial step in overcoming the data scarcity challenge. They contribute to the advancement of autonomous farming technologies and are already attracting significant international attention.

Each dataset features temporally sparse annotations, meaning that the annotations for one image do not overlap with those of other images. The captured field is separated into three herbicide treatment regimes (30%, 70%, 100%) which provide a large range of distribution variations of different classes. For every plant instance in these datasets, annotations include the stem point, pixel-wise segmentation, and bounding box, along with both super and subcategories identified with expert assistance.

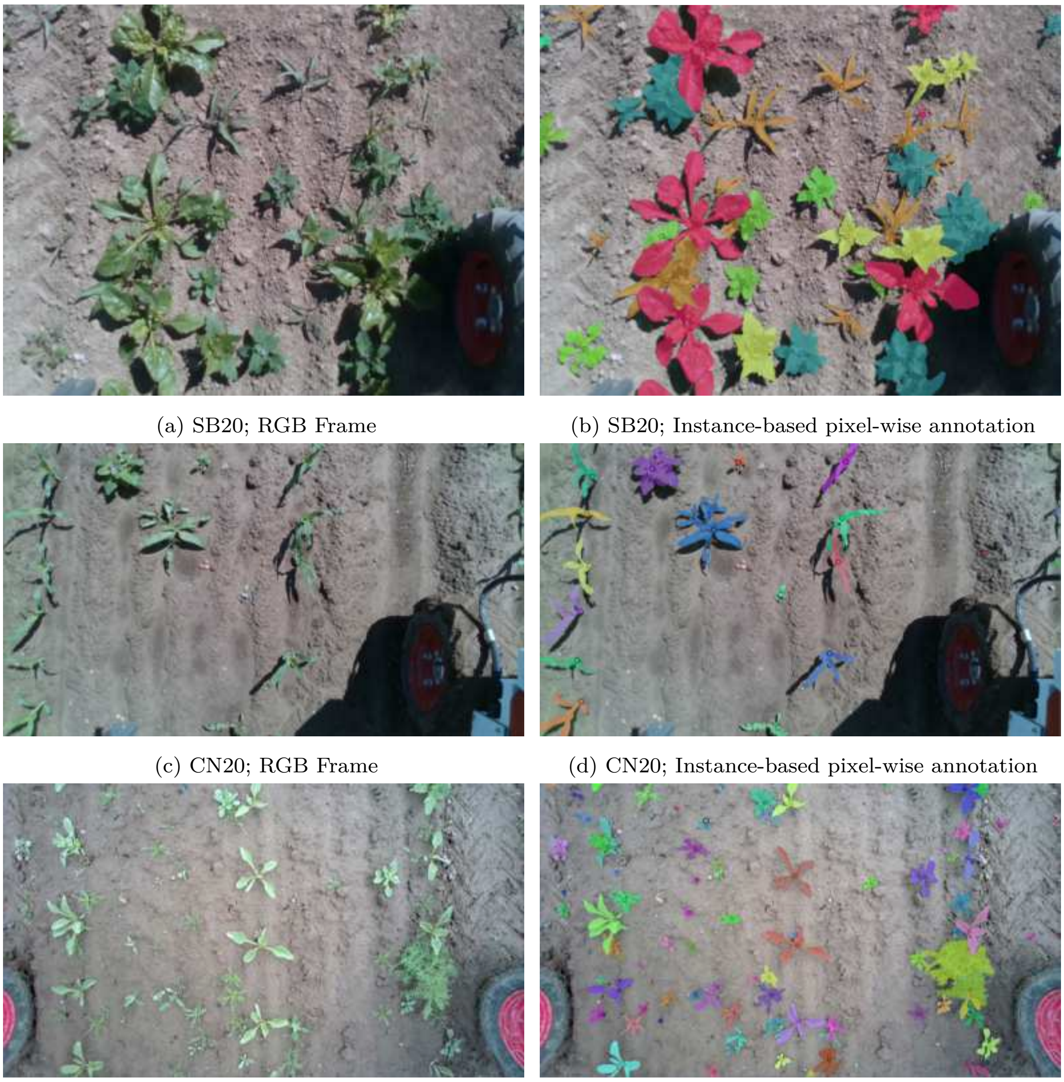
Additionally, every image in the dataset is geo-referenced and accompanied by relevant robot position, orientation, velocity, raw odometry, IMU, and control data in Coco format. These datasets contain RGB-D images of crops (SB20 and SB21: sugar beet, CN20: corn) and eleven different categories of weeds. In each dataset, several variations occur naturally, including, weed species, crop and weed densities,arious ranges of growth stages, illumination variation, and shadowing caused by the platform. These datasets represent 21271 instances of the plant (see Table 3.3) through 675 RGB images (with available Depth, IR, location, orientation, robot metadata, etc), annotated manually in almost 520 hours.

The distribution of plants and instances is shown in Table 3.5, also Figure 3.12 shows an example set of original and annotated images of SB20, CN20, and SB21 datasets.

The datasets were created using Coco-Annotator, an open-source tool that provided JSON annotations. Metadata, covering training, evaluation, and validation set details, along with camera and robot capture parameters, was stored in YAML files. The data was initially acquired in ROS bag file format, requiring additional steps to produce the final dataset. This included sub-sampling and synchronizing data for a training subset with labeled instance segmentation

Table 3.2: Dataset's Image and meta-data Characteristics; Summary of image count, resolution, frame rate (Fps), channels, and sensor data (IMU, odometry, GPS) for the SB20, CN20, SB21, SB22, and MC22 datasets.

Name	Image Num.	**Resolution**	**Fps**	**Channels**	**IMU**	**Odometry**	**GPS**
SB20	143	640 × 480	30	RGB-D-IR	✓	✓	✓
CN20	283	1280 × 720	15	RGB-D-IR	✓	✓	✓
SB21	84	1280 × 720	15	RGB-D-IR	✓	✓	✓
SB22	20	1280 × 720	15	RGB-D-IR	✓	✓	✓
MC22	145	1280 × 720	15	RGB-D-IR	✓	✓	✓



(a) SB20; RGB Frame

(b) SB20; Instance-based pixel-wise annotation

(c) CN20; RGB Frame

(d) CN20; Instance-based pixel-wise annotation

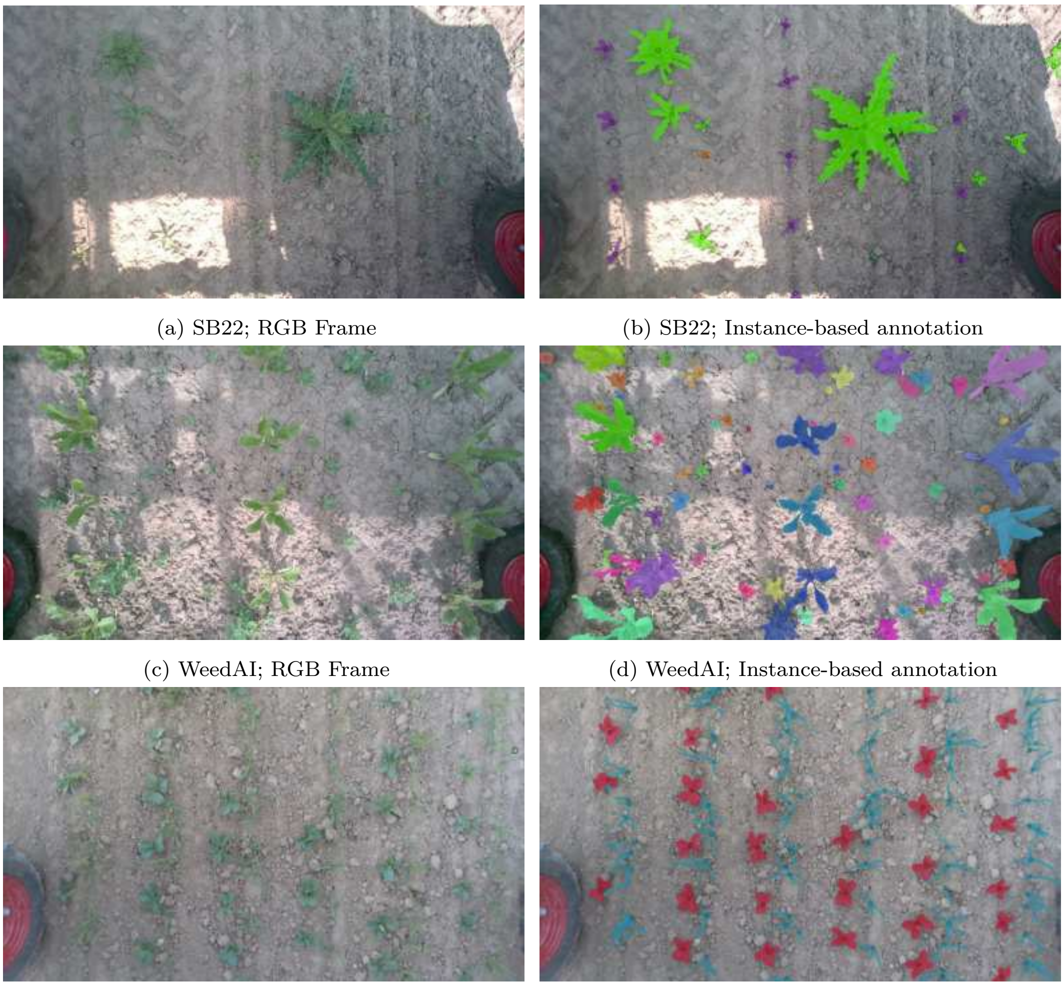
(e) SB21; RGB Frame

(f) SB21; Instance-based pixel-wise annotation

Figure 3.11: Example image of datasets; (top)*SB20*, (middle) *CN20*, (bottom) *SB21*. On the right, the raw RGB dataset is displayed, while on the left, multi-class annotations are used to represent different crop and weed types, each distinguished by a specific color. In the SB20 (top) sample visualization, annotations are color-coded based on class IDs: all crops are shown in red, while each weed class is marked with a unique color. In contrast, the visualizations for CN20 (middle) and SB21 (bottom) use instance-based annotations, where each plant is assigned a different color.

annotations.

The recorded bag files contain various system topics, such as camera data and robot motion parameters, where each was sampled at different frequencies. To address this, an initial synchronization of key topics was performed at the device level. The Figure 3.13 illustrates our approach for aligning timestamps, where data was capped at the lowest frame rate (15Hz, for camera RGB and depth images) to achieve synchronization. Afterward, we sub-sampled the extracted



(a) SB22; RGB Frame

(b) SB22; Instance-based annotation

(c) WeedAI; RGB Frame

(d) WeedAI; Instance-based annotation

(e) Mixed-Crop 22; RGB Frame

(f) Mixed-Crop 22; Instance-based annotation

Figure 3.12: Example image of datasets; (top)*SB22*, (middle) *WeedAI*, and (bottom) *Mixed-Crop 22*. On the right, the raw RGB dataset is shown, while on the left, multi-class annotations are applied to represent various crop and weed types, with each type identified by a unique color.

Table 3.3: Dataset's Annotation Characteristics; Overview of annotated instances, annotation time, and the number of samples in the training, validation, and evaluation sets for SB20, CN20, SB21, SB22, and MC20 datasets.

Name	**Ann. Instances**	**Ann. Time**	**Train Set**	**Valid Set**	**Eval Set**
SB20	2925	35.1 h	71	37	35
CN20	2566	93.7 h	149	39	44
SB21	5931	162.3 h	56	14	14
SB22	282	8.0 h	-	-	-
MC22	9849	221.3 h	-	-	-

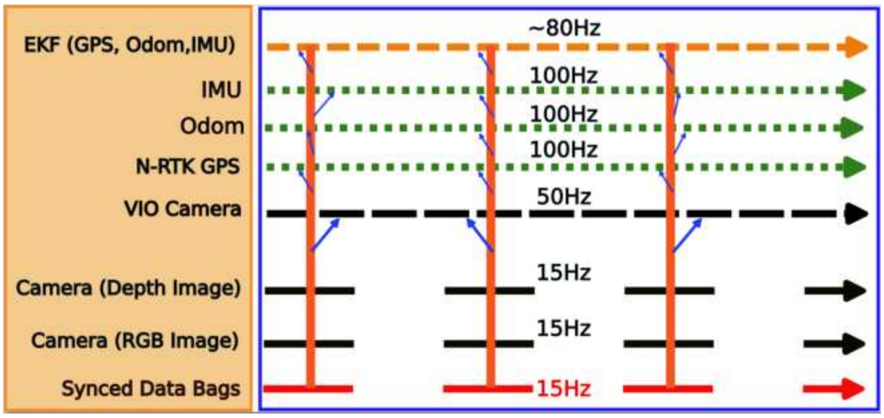


Figure 3.13: Data synchronization architecture for BonnBot-I sensors; where various sensor data streamsranging from high-frequency IMU, odometry, and GPS data to lower-frequency camera imagesare aligned based on the lowest frequency source (camera images at 15Hz) to ensure synchronized data logging (length of dash-lines indicates how long data acquisition takes).

images to avoid overlap and matched each image with associated GPS, IMU, odometry, and other sensor data.

3.4 Crop and Weed Monitoring

Agricultural monitoring application is widely used in different cultivars and farming sites including in glasshouses [101, 99], orchards [110], and fields (for weed intervention) [78]. As discussed previously, one of the primary capabilities of an autonomous precision agricultural robotic platform is its ability to monitor the field in real-time as it traverses, complementing its core function as a weeding system. Similarly, BonnBot-I relies on a monitoring approach that utilizes Mask-RCNN for instance segmentation, providing species-level identification (e.g., crop and weed) and surface area estimation. We introduced the main method in [52] where our dynamic radius (DR) spatial matching operator as a tracking-via-segmentation method outperforms the traditional intersection over union (IoU) method. The technique also leverages re-projection [100] between frames using wheel odometry and camera parameters supplied by BonnBot-I.

In the original work [52] we only relied on wheel odometry data of BonnBot-I to utilize the re-projection technique. However, wheel odometry is susceptible to errors, making it less reliable. Hence in [5], we showcase BonnBot-I's potential for field monitoring by enhancing its tracking capabilities through the integration of additional localization sensors and utilizing extended Kalman filter-based (EKF) Section 2.1.3 sensor fusion.

In the following, we elaborate on the details of our approach for detecting each plant underneath the robot and tracking them to enable precise intervention. Which is followed by a subsection of experiments using our monitoring system.

Finally, by relying on the result of the monitoring system, we introduce our method for creating reusable field models based on real observations and the output of the monitoring system to enable the development and evaluation of weeding algorithms. However, we should note that developing a novel vision system was not the main focus of this thesis, therefore we will briefly discuss developed improvements and strategies empowering targeted intervention.

3.4.1 Instance-Based Semantic Segmentation

In [52], we introduced a DNN-based monitoring approach designed to be both crop- and robot-agnostic. Figure 3.14 provides an overview of the introduced monitoring approach, with extensions for re-projection tracking, implemented on PATHoBot [100] and BonnBot-I [5] for horticultural and arable farming applications.

Our vision pipeline is consistent across both robots, differing only in the deep-learning model, each tailored to specific sub-classes. In this setup, the robot scans the environment while executing instance segmentation, enabling accurate localization of plants/fruits along with area estimations and classification within the scene. To achieve this, we utilize instance segmentation masks from Mask-RCNN [54] as the base network. Our instance-based segmentation approach segments each plant individually, enabling unique data points for every object.

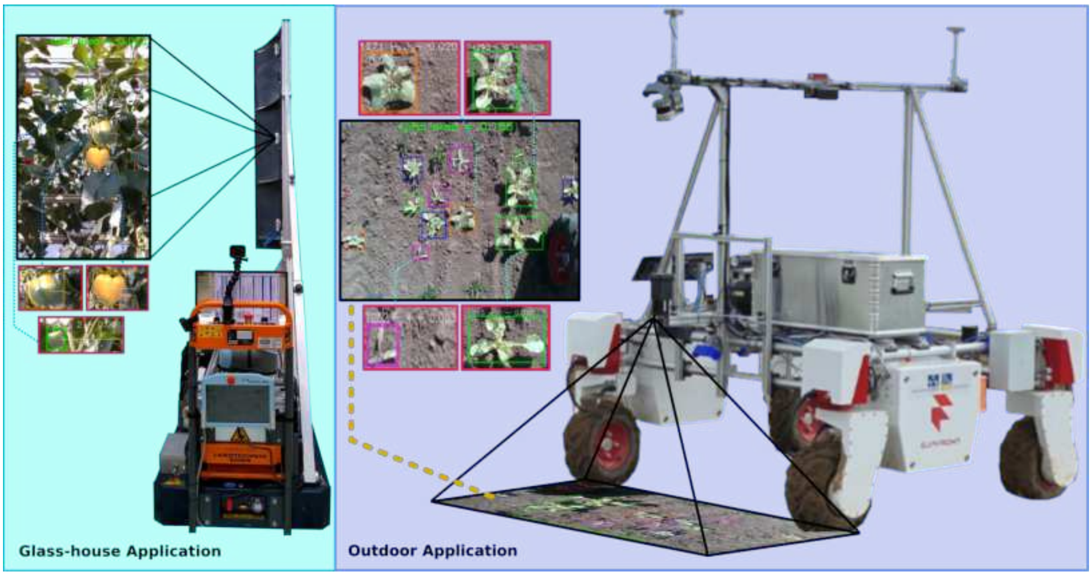


Figure 3.14: The agnostic monitoring algorithm provides up-to-date information to the farmer based on instance segmentation with ripeness or species information and area estimation. This assists in making more informed management decisions such as weeding or harvesting using a tracking-via-segmentation approach for yield estimation. The approach is evaluated on two robotic platforms PATHoBot (left) and BonnBot-I (right) which work in significantly different environments: glasshouse or arable fields [52].

The super-class labels distinguish between background (soil) and plant. The sub-class classification layer further refines this by providing quality metrics, such as identifying specific weed species in arable farmland. As demonstrated in [53, 51], this sub-classification method improves both classification and segmentation accuracy without losing information.

Figure 3.15 details the super- and sub-class layout. This approach leverages flexible super- and sub-class relationships for enhanced classification beyond traditional multi-class systems, where class quantity or quality constraints could degrade performance.

Following up this instance-based perception, the predictions are aggregated through a tracking-via-segmentation method resulting in unique-per-target tracklets. Finally, tracklets information is processed to generate key data, supporting intervention planning for subsequent manual or robot-based operations.

3.4.2 Estimating the Area

For informed intervention decisions, object quality or species alone is insufficient. Area estimation adds a critical information layer relevant to weeding and harvesting. This supports better decisions for crop management, labor allocation, and business planning (e.g., abundant, ripe sweet peppers indicate readiness for harvest).

Phenotypic information is essential for informed decisions on monitored objects. In arable farmland, plant area estimation provides insights into the growth stage. Likewise, weed presence alone doesnt justify herbicide use; weed count and area offer biodiversity data that can influence treatment decisions. To estimate object area, we leverage the stereo vision capabilities of the sensors on the BonnBot-I platform. The area is calculated using registered depth and camera focal length following object segmentation. Once a masked area is identified,

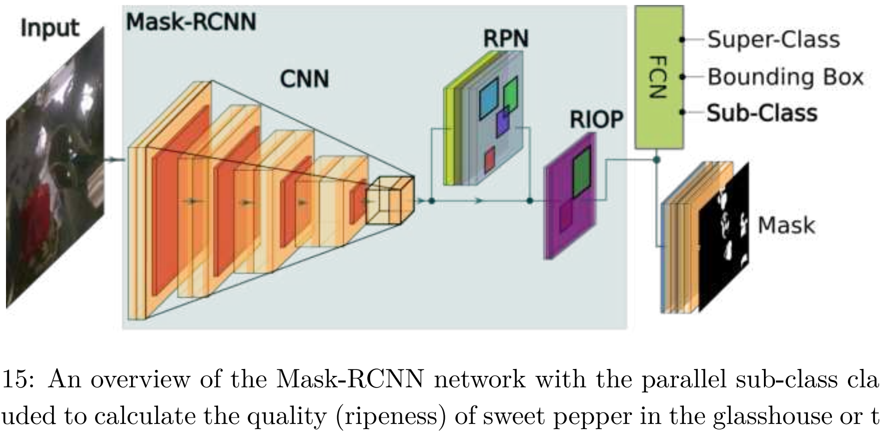


Figure 3.15: An overview of the Mask-RCNN network with the parallel sub-class classification layer included to calculate the quality (ripeness) of sweet pepper in the glasshouse or the species of crops/weeds in arable farmland [52].

the area (A) of the m-th object is calculated as follows:

$$A_m = \sum_{i}^{N} \frac{d_i^2}{f_x \times f_y} \tag{3.1}$$

The N represents the total number of pixels in the segmented region (1D vector), and depth values (d, in meters) come from the registered depth image. f_x and f_y are the cameras focal length parameters, sourced from its intrinsic properties. With this sensor framework, we provide richer farm information to end users. No prior scene scaling assumptions are needed, as the sensor supplies pixel-wise depth, enabling direct area calculation for crop surfaces.

Adding area as a metric to segmentation and classification (super- and sub-class) enhances data on the objects and entire rows being scanned. This supports tracking plant growth time, aiding in farm management. For weeding and crop management, area estimation enables more precise interventions (e.g., large crops but small weeds imply no immediate weeding is needed). Integrating these metrics into intervention decisions provides a more robust decision-making framework, though currently based on single-image analysis.

3.4.3 Tracking-via-segmentation

As discussed previously, one major part of our monitoring and environmental perception software is tracking individual crops in a field which is crucial for accurate and targeted interventions or providing useful monitoring information. In agricultural environments, the scenes relative stability and predictable robot movement support tracking by segmentation. Tracking-via-segmentation takes advantage of the relatively stable agricultural scene, where plants remain spatially static as the robot moves along a row. This allows us to assume that objects imaged at time t will appear nearby in position at $t + 1$, given the platform's moderate speed and camera frame rate.

An initial approach by [53] used tracking-via-detection with bounding boxes, such early tracking approaches relied heavily on spatial and temporal consistency, yet were prone to duplicate tracklets when image captures were too far apart. later expansions to tracking-via-segmentation with reprojection by [100] showed that by using odometry data and depth information, reprojection could compensate for larger robot movements between frames, improving tracklet accuracy. However, small discrepancies in captures posed a challenge for IoU-based matching, particularly for small objects, as noted in [52].

To address this, [52] introduced a dynamic radius criterion, which compares the Euclidean distance between object centers rather than relying on IoU alone. If the center distance is within a threshold radius, the objects are merged into a single tracklet. While effective, this approach occasionally mismatches due to its 360-degree matching around the object center.

By combining mask output from a deep-learning classifier with depth and odometry data, we can predict object location from frame $t - 1$ to t, forming the basis of reprojection for tracking [100]. The tracklets act as the baseline of the approach to maintain the identity of an object and aggregate the instance segmentation as the robot traverses the scene. Tracklets maintain each crop's identity as the robot moves through the field, aggregating segmentation data. Using a greedy algorithm, tracklets at $t - 1$ are matched to new masks at t, updating tracklets if a match is found or creating new tracklets if not.

The position of an active tracker at t is updated with the position of the detection with the highest IoU value at $t + 1$ if the IoU value is greater than a threshold (0.7). Otherwise, the track is not considered active, and if the inactivity happens for several frames (3), the track is removed from the list of active tracks, then the number of plants is added by one if it satisfies the minimum number of tracks. For the sub-class of the plant, the most frequent prediction is taken for the output sub-class.

To ensure robust matching, only objects fully visible within the image bounds are tracked, reserving entry and exit zones per input image [100]. Matching criteria are key to effective tracking, so four methods are evaluated: IoU and dynamic radius, each with and without reprojection. The IoU metric compares tracklet and mask shapes based on overlap, while dynamic radius calculates center distance, especially benefiting small object tracking. Each approach leverages reprojection to further stabilize tracking through varying scene conditions.

3.4.4 Dynamic Radius

The dynamic radius (DR) metric addresses the limitations of pixel-wise intersection over union (IoU) by representing each object as a center point with a radius proportional to its size. This approach removes the need for precise alignment required by IoU. A comparison of DR and IoU is shown in Figure 3.16 with both small (bottom row) and large (top row) objects; the smaller plant is zoomed in for clarity. The final images in Figure 3.16 illustrate how DR accommodates a slight shift of five pixels in each direction. In this example, the IoU method (second from the right) fails to maintain significant overlap for the smaller plant, while DR (rightmost image) still effectively includes the shifted location within its matching region. For larger objects, the flexible DR boundaries simplify matching compared to the strict IoU requirement.

The DR metric begins by calculating the center of mass (mean of x and y samples) for each object, as shown in Figure 3.16. Then, the DR is set to the greatest distance from the center to the bounding box edge along either the x or

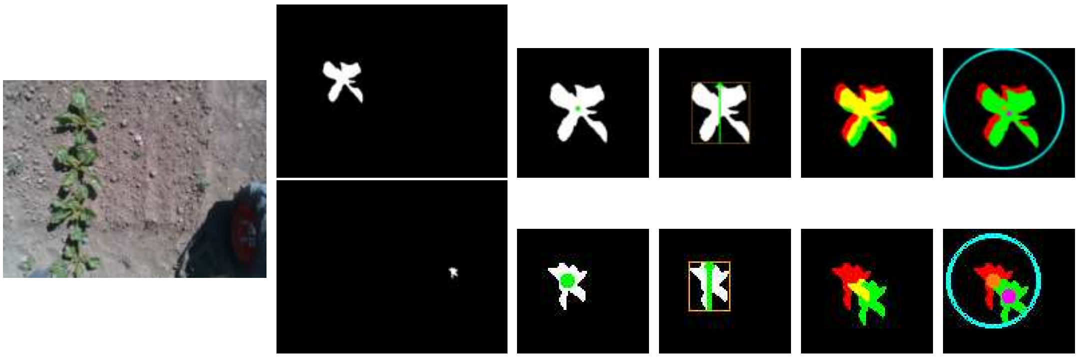


Figure 3.16: The dynamic radius calculation, far left is the original RGB image where we select two of the plants in the scene. The top row is a large crop example and the bottom row is a small weed. From left to right (after the RGB) the segmentation map, finding the center of mass location, calculating the radius of the search, the IoU after a small shift (5 pixels in each direction), and finally the centroid locations and the search radius. The bottom row has been scaled up in resolution to match the top row and is a considerably smaller plant [52].

y direction. Using this radius, we apply an Euclidean distance filter to exclude objects outside this radius from being matched. For tracking, a masks center of mass closest to the tracklet and within the DR is matched.

3.4.5 Re-projection

In both IoU and DR metrics, limited spatial shifts between frames are crucial. To address this, [100] introduced reprojection, allowing the alignment of tracklets between consecutive frames. This reprojection approach increased tracking performance in cluttered weeded scenes by adjusting the location of previous tracklet masks to newly segmented regions. Reprojection uses motion information of BonnBot-I to compute the camera transformation between frames i and j:

$$\mathbf{H}_{ij} = \mathbf{E}^{-1}\mathbf{W}_{ij}\mathbf{E}, \tag{3.2}$$

where $\mathbf{W}$ and $\mathbf{E}$ represent the motion transformation and camera extrinsics. Using this transform, pixel coordinates $\mathbf{m}$ in the detection mask $\mathcal{M}_i$ from frame i are projected to frame j as:

$$\mathbf{m}_{jk} = \pi(\mathbf{H}_{ij}(\pi^{-1}(\mathbf{m}_{ik}, d_{m_{ik}}))), \tag{3.3}$$

where $k = [1, \ldots, N]$, $\pi(.)$ is the camera projection function, and d_m is each coordinates depth.

3.5 Experiments

This section presents an experimental evaluation of our field monitoring approach, deployed on our self-developed robot BonnBot-I. Our evaluation covers various components of the proposed method, starting with the performance assessment of the instance-based semantic segmentation algorithm, which includes crop detection and sub-class classification accuracy. We then analyze the effectiveness of our tracking methods, both with and without reprojection, emphasizing the robustness of the reprojection approach amid significant frame skips. Furthermore, examining the impact of sensor fusion on localization accuracy and tracking performance. Additionally, we introduce high-fidelity field models to assess weeding methods across varied weed densities and field arrangements.

3.5.1 Experimental Setup

We implemented the Mask-RCNN using PyTorch and trained it for 500 epochs with a 0.001 learning rate using stochastic gradient descent. For datasets SB21, SB22, and WeedAI, images are resized to 704 × 416 as in [4]; SB20 maintains its full resolution of 640 × 480. The training uses a batch size of six, and the model is selected based on validation set performance, which may occur before the 500th epoch. Key hyperparameters include a learning rate of 0.001. For non-maximum suppression and IoU matching, we use thresholds of 0.4 and 0.1, as established in [51] and [53]. In the following, our evaluation, methodology involves three stages: (1) assessing object detection accuracy, (2) evaluating segmentation quality, (3) analyzing the precision of the sub-class layer, and (4) evaluation of tracking performances on similar datasets.

3.5.2 Object detection

The object detection accuracy is quantified using localization metrics, where a true detection (T_P) is identified by sufficient overlap with ground-truth positions at an IoU threshold of 0.1. On the SB20 dataset, our approach achieves Precision, Recall, and F1-Score values of 0.89%, 0.85%, and 0.87%, respectively. For the CN20 dataset, these values improve to 0.96%, 0.88%, and 0.92%, while on the SB21 dataset, the approach yields Precision, Recall, and F1-Score values of 0.96%, 0.72%, and 0.82%, respectively.

3.5.3 Instance- and Semantic Segmentation

Instance segmentation extends object detection by accurately identifying all pixels corresponding to an object. Instance-based segmentation forms the foundation of our proposed approach, utilizing Mask-RCNN enhanced with a novel parallel layer for sub-class classification. This parallel structure allows the model to simultaneously learn broad (super-class) and specific (sub-class) categories.

IoU serves as the primary evaluation metric, though it tends to show lower scores due to the impact of false positives and missed detections. By comparing Mask-RCNN outputs with ground-truth masks, we achieve detailed performance insights. Table 3.4 presents the instance and semantic segmentation results for the SB20, CN20, and SB21 datasets.

Table 3.4: Instance and semantic segmentation results for the datasets. Metrics include background IoU (BG IoU), foreground IoU (FG IoU), and average IoU (μIoU).

	Instance Segmentation			Semantic Segmentation		
Dataset	BG IoU	FG IoU	μIoU	BG IoU	FG IoU	μIoU
SB20	0.99	0.49	0.74	0.97	0.72	0.85
CN20	0.99	0.61	0.80	0.99	0.74	0.86
SB21	0.99	0.48	0.74	0.97	0.67	0.82

The results highlight the impact of false and missed detections on instance segmentation performance. For the SB20 dataset, an FG IoU of 0.49% illustrates that while segmentation accuracy is high for correctly predicted regions, false and missed detections considerably influence the overall IoU score. To better understand these effects, we also evaluated a simplified two-class semantic segmentation scenario (plant vs. background), where the influence of false and missed detections is reduced. As shown in Table 3.4, semantic segmentation delivers robust FG IoU performance, closely aligned with results obtained by excluding false and missed detections from instance segmentation. This demonstrates that, despite challenges, our parallel layer-enhanced Mask-RCNN effectively segments images based on object properties, providing a reliable approach for field-based plant analysis.

3.5.4 Sub-class Accuracy

To effectively implement plant-specific interventions, accurate classification of both crops and weed species is essential. This task is challenging due to the diverse growth patterns and varying, unconstrained shapes of different species. Our evaluation across multiple datasets SB20, CN20, and SB21reveals significant variations in classification accuracy among species, often correlating with the number of training samples available. So we group different species in three different groups based on prediction accuracy in blow:

- **High Accuracy Species:**

  Crop (Sugar Beet (SB) and Cron (CN)): Consistently high accuracy across all datasets (SB20: 0.946, CN20: 0.988, SB21: 1.000), reflecting the abundance of training samples and the importance of precise crop identification in agricultural management. *Persicaria lapathifolia (Pe)*: Achieved high accuracy in SB20 (0.990) and SB21 (0.917), indicating effective classification when sufficient training data is available. *Lamium purpureum (Lp)*: Present only in SB21, with perfect accuracy (1.000), suggesting that adequate representation in the training set enhances classification performance.

- **Moderate Accuracy Species:**

  Bidens pilosa (Bi): Displayed moderate accuracy in SB20 (0.615) and CN20 (0.727), but was absent in SB21, indicating variability possibly due to differing sample sizes or dataset characteristics. *Thlaspi arvense (Th)*: Showed moderate accuracy in SB20 (0.731) and SB21 (0.907), with absence in CN20, suggesting that dataset-specific factors influence classification performance. *Chenopodium hybridum (Ch)*: Recorded moderate accuracy in SB20 (0.792), CN20 (0.806), and SB21 (0.906), indicating relatively consistent classification across datasets. *Veronica persica (Ve)*: Present only in SB21 with an accuracy of 0.909, suggesting effective classification when included in the dataset. *Polygonum aviculare (Po)*: Achieved perfect accuracy in CN20 (1.000) and high accuracy in SB21 (0.889), indicating reliable classification in these datasets. *Solanum nigrum (So)*: Present only in SB21 with an accuracy of 0.500, suggesting challenges in classification, potentially due to limited training samples.

- **Low Accuracy Species:**

  Anthemis arvensis (An): Exhibited low accuracy in SB20 (0.333) and SB21 (0.953), with perfect accuracy in CN20 (1.000), indicating inconsistencies possibly due to varying sample sizes or dataset characteristics. *Chenopodium album (Cy)*: Showed low accuracy in SB20 (0.475) and was absent in CN20 and SB21, suggesting challenges in classification, potentially due to limited representation in the training data.

These findings underscore the importance of balanced datasets with sufficient sample sizes to achieve reliable classification across diverse plant species. The high accuracy observed for Sugar Beet (SB) is particularly valuable, as precision in crop identification is crucial for effective crop monitoring and precision weeding, where misclassification of weeds is less critical. In summary, combining precise pixel-wise object localization with robust species classification provides farmers with valuable phenotypic insights, enhancing decision-making in agricultural management.

Table 3.5: Classification Accuracies Across Datasets; This table presents the classification accuracies for various plant species across three datasets: SB20, CN20, and SB21. Each cell indicates the accuracy achieved for a specific species within a particular dataset, with dashes representing species not present in the evaluation test. Higher accuracy values denote better classification performance, which is crucial for effective plant-specific interventions in agricultural settings (Note that '-' indicates the species was not present in the evaluation test for that dataset).

Name	**Crop**	**Bi**	**An**	**Cy**	**Pe**	**Th**	**Ch**	**Ve**	**Po**	**Lp**	**So**	unknown
SB20	0.946	0.615	0.333	0.475	0.990	0.731	0.792	-	-	-	-	0.368
CN20	0.988	0.727	1.000	0.000	0.000	0.727	0.806	0.000	1.000	-	-	0.000
SB21	1.000	0.000	0.953	0.000	0.917	0.907	0.906	0.909	0.889	1.000	0.500	0.826

3.5.5 Tracking Evaluation

Effective tracking is essential for developing robust agricultural monitoring algorithms, as it enables precise in-field interventions by providing detailed phenotypic data, such as plant species and growth stages. To achieve this, specific tracking parameters are employed. The "keep running" parameter allows a tracklet to skip up to 5 frames, accommodating temporary occlusions or missed detections. A tracklet must have a minimum of 10 segmentation matches to be considered valid, ensuring reliability in tracking data. An Intersection over Union (IoU) threshold of 0.1 is used to associate tracklets with new regions, balancing sensitivity and specificity in matching. Empirical testing of weighting values (0.5, 0.75, and 1.0) for the Dynamic Radius found 1.0 to be optimal, enhancing tracking performance. To prepare ground truth for tracking experiments, the evaluation rows of the datasets are manually annotated (crops and weeds are counted manually), to ensure accurate and reliable data for performance assessment.

All datasets feature objects solely on the ground plane, eliminating the need for depth filtering and simplifying tracking evaluation. Table 3.6 presents tracking results across three datasets (SB20, CN20, and SB21), evaluated using different criteria: IoU, IoU with Reprojection, DR, and DR with Reprojection (introduced in Section 3.4.5).

In agricultural monitoring systems, tracking algorithms are evaluated using metrics such as the coefficient of determination (R^2) and mean normalized absolute error (μNAE). As the first observation from the results in Table 3.6, high R^2 values more than 0.93 across various datasets indicate a strong linear correlation between predictions and ground truth, underscoring the reliability of these tracking methods.

The SB20 dataset, characterized by lower-resolution imagery and a high diversity of plant and weed species at various growth stages, presents challenges due to the complexity and variability of visual data. In this context, the Dynamic Ra-

Table 3.6: Tracking-via-Segmentation results using different criteria across datasets (SB20, CN20, and SB21). Results display the R^2 value and the mean normalized absolute error (μNAE).

	SB20		CN20		SB21	
Approach	R^2	μNAE	R^2	μNAE	R^2	μNAE
IoU	0.937	0.392	0.997	0.717	1.000	0.328
IoU Reproj	0.947	0.269	0.975	0.257	1.000	0.411
DR	0.957	0.208	0.993	0.083	1.000	0.083
DR Reproj	0.970	0.131	0.962	0.035	1.000	0.034

dius (DR) approach with reprojection achieved a μNAE of 0.131, outperforming the IoU with reprojection method, which had a μNAE of 0.269.

Conversely, the CN20 dataset comprises cleaner fields with low weed density and limited weed diversity, primarily featuring weeds at small growth stages. This simpler environment contributes to higher R^2 values and lower μNAE in certain approaches. For instance, in CN20, the DR trackers μNAE improved from 0.083 to 0.035 with sensor fusion and projection technique, effectively halving the error. Similarly, the IoU trackers μNAE decreased from 0.717 to 0.257 with enhanced localization, demonstrating the benefits of integrating multiple localization sensors and sensor fusion techniques in improving tracking accuracy.

The SB21 dataset presents a scenario with very high weed density, exceeding 70 weeds per square meter, diverse weed growth stages, and limited weed diversity, posing significant challenges for tracking algorithms due to the dense and heterogeneous plant distribution. In SB21, all tracking approaches achieved an R^2 value of 1.000, indicating a perfect linear correlation between predictions and ground truth. The μNAE varied across methods, with IoU with reprojection exhibiting the highest error at 0.411, while DR with reprojection achieved the lowest error of 0.034. These results demonstrate that the Dynamic Radius approach, particularly when combined with reprojection, outperforms IoU-based methods in tracking accuracy within the SB21 dataset.

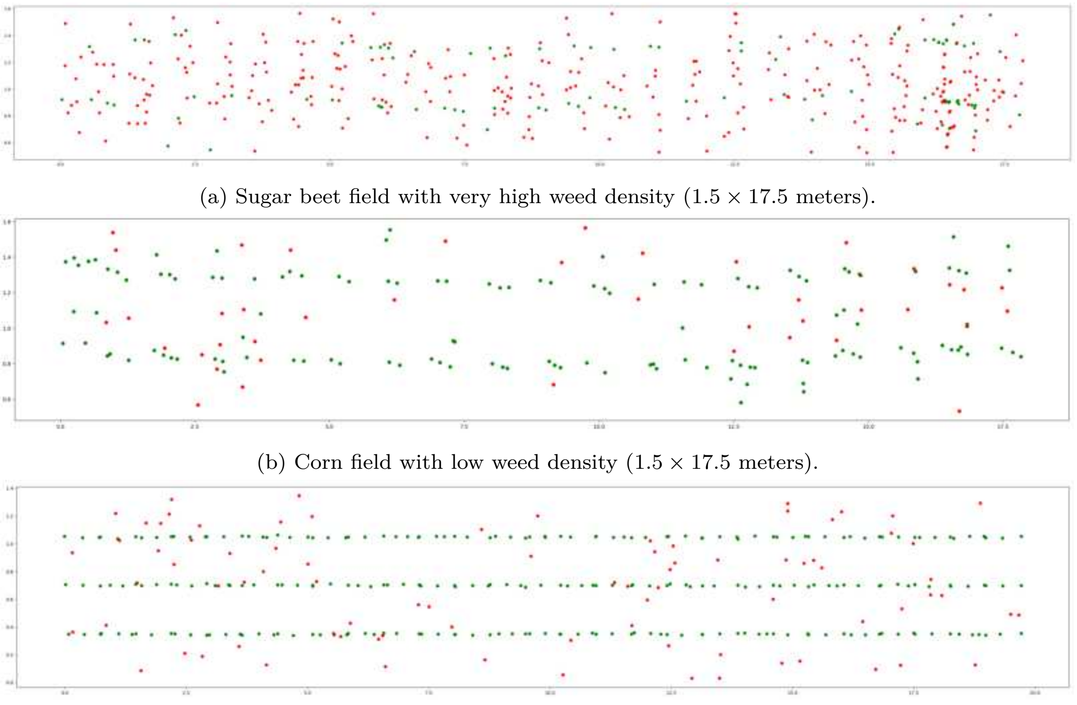
These varying characteristics across datasets highlight the adaptability and robustness of the Dynamic Radius approach with reprojection, as it consistently delivers superior tracking performance despite differences in image resolution, plant diversity, and weed density.

3.5.6 Field Models

Field models of sugar-beet and corn crops were developed as part of an innovative crop monitoring approach. All models captured real-time system performance at the CKA campus over multiple years.

Our novel method compiles crop-specific data from real-world fields, to create high-fidelity models representing varying weed densities and growth stages. The models organize complex agricultural data into a structured row format that simulates real-field conditions for advanced performance analysis.

A major achievement of this work is the representation of four distinct weed densities: low (CN20), moderate (SB20-S2), high (SB20-S1, SB21-S1), and very high (SB21-S2). This diversity allows for realistic test scenarios that capture the large variability in weed populations, as shown in Figure 3.17. Evaluating the crop monitoring system with such detailed models enables precise operational efficiency assessment using two key metrics. The first is the 'loss' percentage, which reflects the number of untreated weeds. The second metric is the linear axis travel distance, which provides insights into workload balance and planning efficiency. This approach contributes to more robust testing in both simulations and real fields. It also sets a new standard for validating crop monitoring technologies, ensuring they are prepared for the complex conditions of real-world agriculture.



(a) Sugar beet field with very high weed density (1.5 × 17.5 meters).

(b) Corn field with low weed density (1.5 × 17.5 meters).

(c) Sugar beet field with moderate weed density (1.5 × 20 meters).

Figure 3.17: Example Field models of sugar-beet and corn crops captured based on the real-time performance of monitoring system Section 3.4 in CKA campus. The moves are gathered and processed throughout four years and include varying weed densities and growth stages.

3.6 Conclusion

In this chapter, we introduced BonnBot-I a transformative advancement in crop monitoring and precision weeding robotic platfrom. During four years of development, we reformed and built the BonnBot-I to be adaptable across various crop types and growth stages and is designed to meet the diverse needs of European phenotyping fields and farm management. We equipped it with a rich sensory network and versatile design that delivers significant advantages in multiple facets of agriculture like autonomous navigation, monitoring, and targeted interventions, making it a promising tool for modern farmers.

BonnBot-I is also capable of performing plant-level interventions on arable farms. This is made possible by its innovative and unique weeding tool, which features several independently controllable weeding nozzles. The system's architecture supports applications on both organic and non-organic farms, offering the ability to use various intervention modalitiessuch as chemical, mechanical, and electricalsimultaneously.

Furthermore, we created a set of specialized public datasets using BonnBot-I, specifically designed to encourage precision plant-level applications. Over four years (2020 to 2023), data was gathered during multiple trial sessions at the University of Bonn's Klein-Altendorf campus (CKA), with a focus on sugar beet, corn, and oat-fava bean (mixed) fields.

We are confident that BonnBot-I has greatly impacted the future of automated precision weeding and intervention in arable farms by facilitating a broad range of applications. These advancements enable farmers to make better-informed weeding and crop management decisions, leading to improved efficiency, sustainability, and higher quality in agricultural production.

Therefore, we conclude that BonnBot-I serves as an excellent platform for real-time precision weeding in various farming environments, effectively addressing our research objectives. In the following chapter, we will investigate incorporating local camera-only observations to guide BonnBot-I through rows of crops in real conditions without relying on any global localization or navigation technologies. This new modality of navigation control aims to enhance autonomous navigation performance and robustness of common problems in a dynamic and challenging environment with cluttered cropping situations and no-GPS-guided seeded farms.

Chapter 4

Vision-Only In-Field Navigation

In this chapter, we present a novel approach to enable autonomous navigation in row-crop fields empowering precision farming and crop monitoring tasks only relying on vision sensors. The proposed approach is originally presented in [3], which exploits the multi-crop row structure using only the local observation from the on-board navigation cameras (introduced in Section 3.1.2) without requiring any global or local position awareness. To achieve this, we have introduced a novel multi-crop row detection strategy that can deal with cluttered and weedy scenes. The presented scheme uses a novel lane-switching strategy that enables BonnBot-I to switch to a new lane of crops (multiple crops underneath the robot) independent of any global positioning system or human intervention. We evaluated our approach on BonnBot-I on up to five crop types (with varying canopy shapes) in real field conditions and three

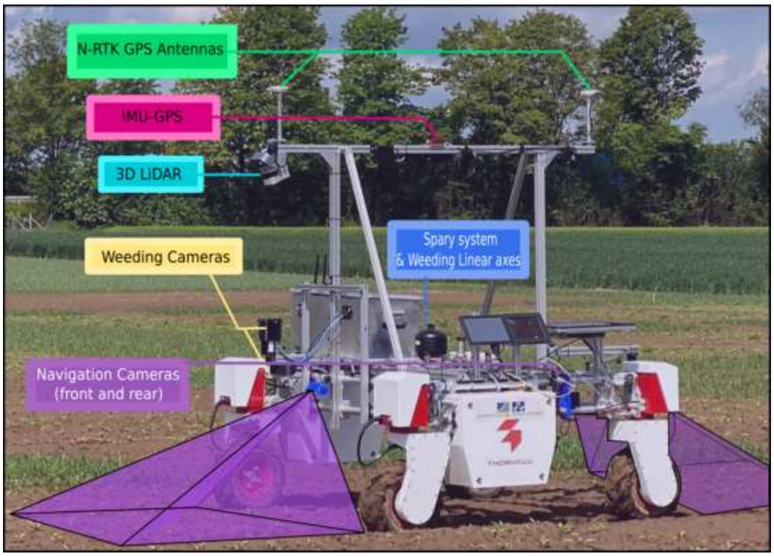


Figure 4.1: BonnBot-I following lanes of crops using two symmetrically mounted monocular cameras in front and back.

challenging simulated fields achieving centimeter-level navigation accuracy in real fields.

Navigating through the field is central to the real-world deployment of agricultural robotics and agricultural vehicles. For autonomous navigation, it is common for platforms to use precise real-time kinematic (RTK) GNSS as it is being used in fully controlled and engineered agricultural sites where they heavily rely on structural information [1, 19]. But, robotic technologies will never be guaranteed accurate GPS in every field, due to its expense and coverage. An example of this is that currently most fields are seeded using traditional methods, not auto-seeding geo-referenced systems, which creates a gap between GPS capabilities and farming requirements [1]. As such, utilizing GPS technology in unregistered fields increases the risk of damaging crops [14]. In a worst-case scenario GPS can fail, therefore, generally applicable crop row following techniques are required.

In our prior work [7] we introduced a GPS-independent technique that was able to traverse a single crop row and switch between adjacent rows. This system was deployed on the skid-street robotic platform and tested using only a single artificial cropping environment. Considering the row-crops are planted in parallel lines, this structure can be exploited to overcome issues such as a lack of germination, which can leave gaps in a crop row, creating navigation issues for single crop row techniques. Such issues can lead to unrecoverable failure cases for a single-crop row navigation system, see Figure 4.2. We greatly extend our prior work [7] and use information from multiple crop rows to navigate a weeding robotic platform in five different real fields under various weather conditions. The proposed approach takes advantage of planting schemes (standardized distances between rows of crops), and ensures minimum damage to crops and persistent coverage throughout the field. The proposed approach relies on well-known signal processing and computer vision approaches, however, their combination is unique and their robustness has been demonstrated by being applied to five crop types on a robot in the field. We consider this approach as an extra navigation controller modality for achieving more reliable traverses in challenging real fields

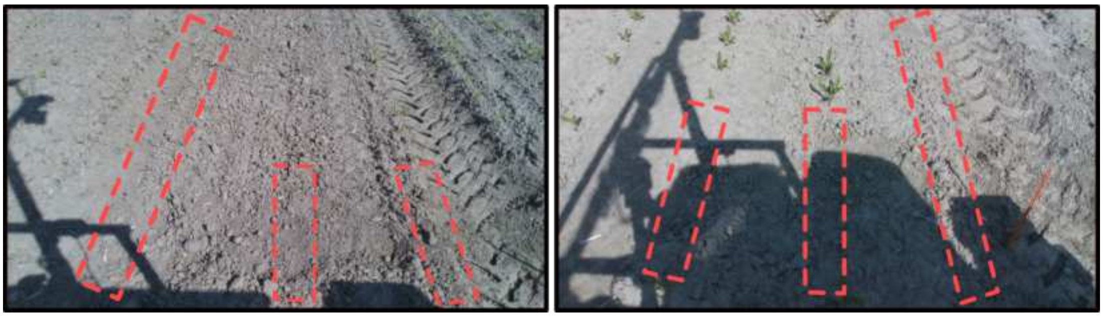


Figure 4.2: Example images where crops have yet to germinate leaving large gaps in a row, indicated by red boxes.

conditions and not denying the potential of benefiting other technologies like GPS and odometry which represents a complete system.

We consider this approach as an extra navigation controller modality for achieving more reliable traverses in challenging fields conditions and not denying the potential of benefiting other technologies like GPS and odometry which represents a complete system. We also made its implementation publicly available [1]. In this chapter, we make the following novel contributions:

- A **robust multi-crop row detection strategy**, capable of dealing with cluttered and weedy scenes for a variety of crop types; and
- A **multiple crop row following approach** which enables autonomous guidance in real-fields; and
- A **multi-crop row lane switching strategy** that enables BonnBot-I to switch to a new lane independent of any global positioning system; and
- **Deployed and evaluated on a real-sized weed management robot** in simulation and the field for five crop types (with different shapes: straight and curved crop rows); and
- **Released a novel multi-row-crop dataset (MultiRowCrop-v1)** covering five crop types with various growth stages under varying illumination conditions.

In the following, initially, Section 4.1 reviews the prior works in subject of autonomous navigation in the row-crop fields and agricultural robotics. Then, we introduce our proposed approach for in-field autonomous navigation in the row-crops field using only local and on-board observations Section 4.2. The Section 4.6 describes the experimental evaluations and implementation details gathered from our simulated and real-field experiments. Finally, the conclusions and future works of this work are drawn in Section 4.7.

4.1 Related Works

Autonomous agricultural robots could improve productivity [113, 25], enable targeted field interventions [77] and facilitate crop monitoring [20]. For uptake of these platforms, they should be deployable in a variety of scenarios including different cultivars, crop row structures, and seeding patterns [112]. A key enabling technology is reliable navigation through the whole field [11].

One potential solution to the navigation problem is to make use of the Global Navigation Satellite System (GNSS). Such an approach has been used for both

[1] `https://github.com/Agricultural-Robotics-Bonn/visual-multi-croprow-navigation`

agricultural machinery [109] and robotic platforms [19]. The downside of this approach is that it relies on an expensive sensor and suffers from limitations such as possible GNSS outages and reliance on geo-referenced auto-seeding. Thus, crop-based navigation techniques leveraging the field structure were investigated for autonomous guidance [21, 7] and in-field interventions [12].

In an attempt to use the structure of a field, Barawid *et al.* [17] investigated LiDAR-based orchards navigation system a similar strategy was used in a simulated environment by [68] for traversing row-crop fields. In addition, Winterhalter *et al.* [117] proposed sensor-independent feature representation from different sensor modalities and detected crop rows. They used LiDAR and camera images to extract single lines in a row-crop field which were spaced equidistantly. While these approaches enable side applications such as obstacle avoidance, frame drift in self-similar environments can cause issues [13], including crop damage.

To avoid crop damage through GNSS or LiDAR failures, RGB based navigation approaches directly exploit the available visual information. These techniques can vary significantly in terms of cost, algorithm simplicity, and availability. Classical machine vision approaches detect crop rows with juvenile crops [12] or detect crop rows under challenging illumination conditions [102].

Kraemer *et al.* [58] used a deep learning approach to reconcile the PSEP features by exploiting the likelihood maps of deep neural networks (DNN). Also utilizing DNNs, [79] proposed a convolutional neural network (CNN) for strawberry crop row detection with accurate navigation. Lin *et al.* [64] also showcases the potential of CNNs to reliably navigate a tea field by classifying tea rows. These approaches are often more accurate than their traditional counterparts for detecting or segmenting specific plants. However, in contrast to traditional approaches, CNNs require a significant amount of labeled data and more computational resources both for training and inference, while being less dynamic and requiring further tuning in different conditions. [35].

To perform vision-based navigation of wheeled platform two common approaches were more popular: proportional-integral-derivative (PID) or visual servoing. Billingsley *et al.* [21] extracted the row of plants using a Hough transform, and used a PID controller for navigation through a sugar-beet field. The visual-servoing [40] technique was also exploited for autonomous car guidance in urban areas [34] by road lane following with obstacle avoidance using a monocular camera. These methods regularize the controlled agent's motion within a loop directly based on the current visual features.

The technique proposed in this chapter draws inspiration from our previous work [7] where we are able to both navigate a single crop row and switch lanes at the end. However, this approach was only tested in a single artificially created row-crop field, without considering real open-field challenges like (different crop types, illumination variation, appearance of weeds, and uneven distribution of plants in rows). We propose a real-field applicable method to automatically detect the number of crop rows. Multi-crop row switching is then enabled by being able to correctly identify new crops as the robot moves across the field.

4.2 In Field Guidance Strategy

A benefit of crop rows is that they are generally planted in consistent long parallel structures. A downside to this parallel structure is that there is no connection between them. Therefore, the platform needs to not only follow the crop row without damaging the crop but also autonomously switch between them.

Note that our navigation scheme allows the robot to transition from one crop row to the next one only by switching the cameras and without requiring the robot to perform a complex maneuver to enter the next row. Furthermore, following our navigation scheme the robot requires a smaller space for maneuvering than the one required to perform a U-turn.

The technique proposed in this section draws inspiration from our previous work [7] where we are able to both navigate a single crop row and switch lanes at the end. However, this approach was only tested in a single artificially created row-crop field, without considering real open-field challenges like (different crop types, illumination variation, appearance of weeds, and uneven distribution of plants in rows). We propose a real-field applicable method to automatically detect the number of crop rows. Multi-crop row switching is then enabled by being able to correctly identify new crops as the robot moves across the field.

We greatly extend our prior work [7] and use information from multiple crop rows to navigate a weeding robotic platform in five different real fields under various weather conditions. The proposed approach takes advantage of planting schemes (standardized distances between rows of crops), and ensures minimum damage to crops and persistent coverage throughout the field. The proposed approach relies on well-known signal processing and computer vision approaches, however, their combination is unique and their robustness has been demonstrated by being applied to five crop types on a robot in the field. To achieve multi-crop row following we employ a similar high-level algorithm to our previous work in [7] for a single crop row. Figure 4.4 outlines our multi-crop row following strategy. Hence, we introduce a visual-based navigation scheme that allows the robot to follow a crop row, exit from it, and enter the next one by exploiting two of the cameras mounted on the front $\mathcal{F}_{C_{\text{front}}}$ and back $\mathcal{F}_{C_{\text{back}}}$ of the robot, as shown in Figure 4.3.

Starting in a corner of the field, The robot finds the number of crop rows underneath it and starts autonomously following the current set of crop rows (a

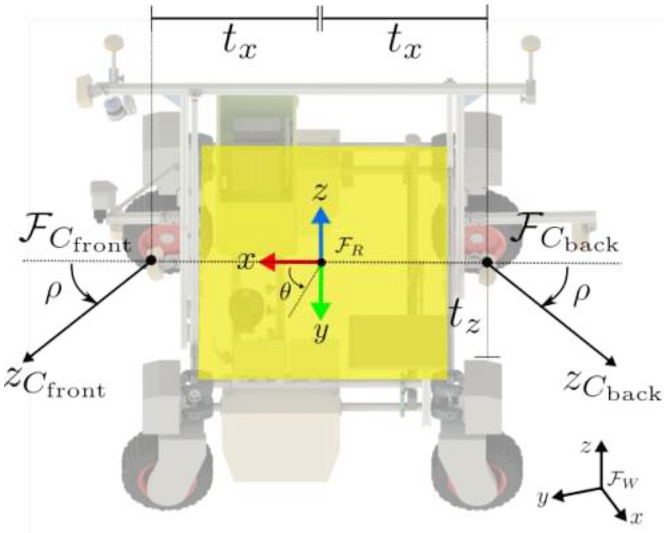


Figure 4.3: Robot, frames and variables; $\mathcal{F}_{C_\text{front}}$, $\mathcal{F}_{C_\text{back}}$ are the front and back camera frames. The cameras are mounted at an offset t_x from the robot center $\mathcal{F}_R$ and t_z above the ground, and with tilt ρ.

lane) using vision-based navigation techniques ① until the front-facing camera detects the end of the current lane. The rear camera then guides the robot to the exit point, end of the lane, actively guiding the robot at all times ②. Using the omnidirectional capability, the robot then switches to the next set of crop rows ③ to be traversed. The benefits of the omnidirectional platform prevail here as we can directly navigate to the new lane without turning around ④, this also outlines the benefit of symmetrically mounted sensors at the front and rear. In the next section we describe the vision-based crop row following and crop row switching algorithms. In the next section we describe the vision-based crop row

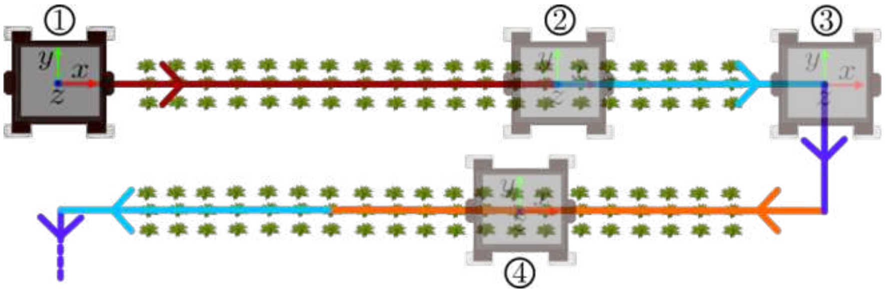


Figure 4.4: In-field navigation strategy, (1) following crop rows with front camera, (2) exiting crop rows using back camera, (3) switching to the next crop rows, and (4) following new crop rows with back camera.

following and crop row switching algorithms. When the robot reaches the end of the crop row, it should exit the current row and enter the next one. However, the camera looking in the navigation direction cannot see any crop anymore. Thus, the robot is still in the row and has no path curve to follow. We introduce a visual-based navigation scheme that allows the robot to exit the current row and enter the next one by exploiting both the cameras mounted on the robot and the symmetry of the robot dynamics.

4.3 Multi-Crop-Row Detection

The first step in a successful vision system capable of traversing a field is the assumption that crop rows are planted parallelly. To have a completely crop-agnostic navigation system the varying distances between the rows for the different crops is an important element. Therefore, it is imperative to have a system that can detect the number of crop rows before instantiating the navigation algorithm.

We perform crop row detection by employing a novel sliding window-based approach. This extracts the location of the prominent crop rows while being robust to the appearance of weeds between them. Our detection approach consists of three steps. First, we perform vegetation segmentation followed by connected components operations to find individual regions (plants) and their center points. This allows us to remain agnostic to the crop that has been planted. Second, we automatically detect the number of crop rows by employing an estimate of the moving variance which we use to describe the field structure. Finally, we track the detected crop rows by centering a parallelogram on each row while the robot is traversing the lane. We detail each of these steps below.

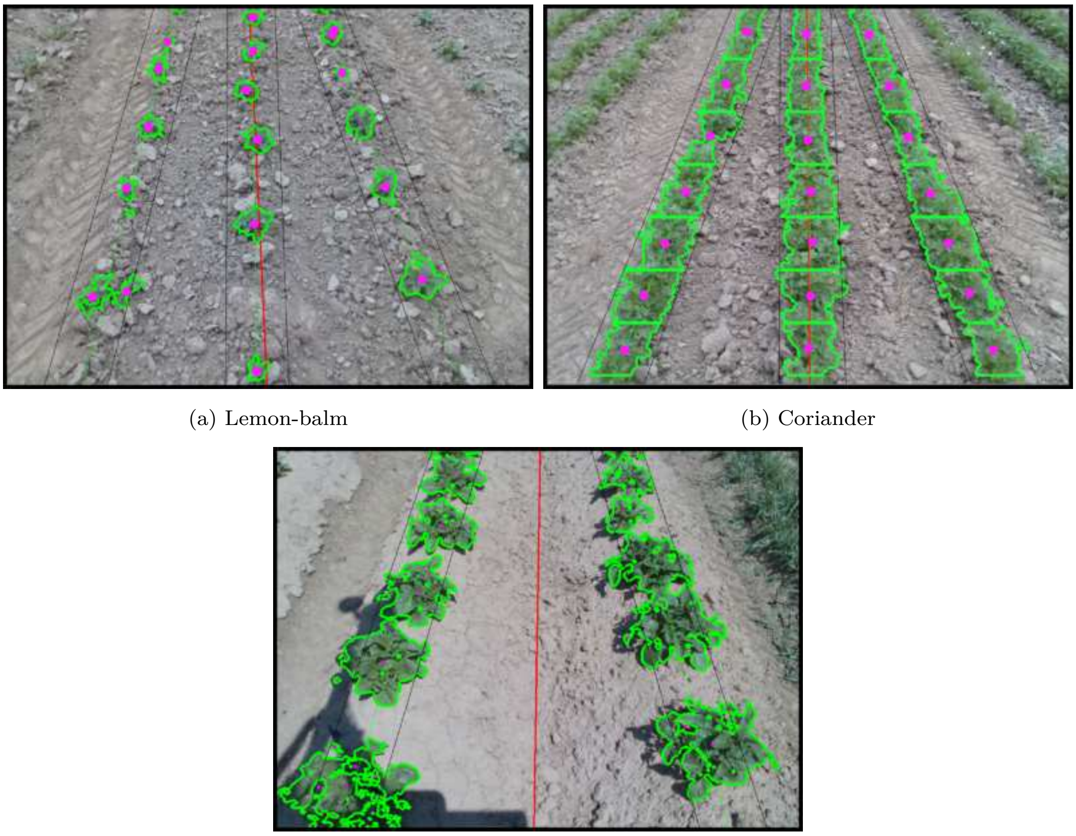
We summarize a row by the position of the individual plants along it. Each plant is represented by its center point. We obtain this by first computing the vegetation mask of the input RGB image using the excess green index (ExG) [118].

$$I_{ExG} \quad = \quad 2I_G - I_R - I_B, \tag{4.1}$$

To separate foreground and background pixels in the image based on ExG we employ Otsu's method [84] which obviates the need for manual tuning of a threshold. Then, each connected component in the vegetation mask is converted to an object of interest (plant) with a unique center point obtained from the center of mass.

One issue associated with this technique occurs when multiple "plants" are absorbed into a single large region most often occurring with bushy plants. A single region representing an entire crop row negatively impacts later stages such as line fitting. To reconcile this we divide contours into smaller subsections if they exceed a predefined maximum height, depicted in Figure 4.5. Ultimately, this step allows us to cater to a larger variety of canopy types.

Figure 4.6 illustrates the crop row detection algorithm, First, a sliding window Ψ scans the image from left to right with a stride of $\mathbf{S}$. The size of the sliding window (w and h) and the stride $\mathbf{S}$ are set to ensure a large overlap between



(a) Lemon-balm

(b) Coriander

(c) Potato

Figure 4.5: The vegetation segmentation (in green), plant boundaries (in bold green), and the resultant plant centers (magenta dots). The location of individual plants is indicated by the enclosed regions (in green), from vegetation segmentation, and their estimated centers are indicated with magenta dots. In (a) individual plants are easy to see (b) is a case where crop boundaries have to be estimated and case (c) is a mixture of both conditions.

adjacent steps in the scans. For the n-th sliding window Ψ_n we compute a line (L_n) based on the crop centers inside the sliding window using the least-squares method. We then find the intersection point I_n of the line L_n with the bottom axis of the image; we only retain lines that intersect within the image bounds. Each line is then described by its point of intersection $I_n = [x, y]$ and the angle of intersection ϕ_n such that $L_n = (I_n, \phi_n)$.

We use the estimated crop lines in conjunction with the moving variance [67] to represent the local structure of the field. The moving variance of the estimated crop line angles, ϕ_n, is calculated over a window of size k such that

$$\sigma^2(\phi_n) = \frac{\sum_{i=n-k/2}^{n+k/2}(\phi_n - \bar{\phi}_n)^2}{k}; \; \bar{\phi}_n = \frac{\sum_{i=n-k/2}^{n+k/2}\phi_i}{k}. \tag{4.2}$$

The moving variance operator yields peaks when there is discord between the local hypothesized crop lines, this occurs between the crop row lines. Troughs occur when there is a consistent agreement regarding the hypothesized crop lines, this occurs in the presence of crop rows. A depiction of this field structure signal is given in Figure 4.6. The peaks (▲) and troughs (▽) of the field structure signal are detected using peak prominence with a constant threshold. To detect the troughs, the signal is flipped (via negation) and peak prominence is applied with the same threshold. The detection of troughs is more complex as crop rows can yield multiple peaks. We resolve this by computing the weighted average of the possible solutions in the local neighborhood, where the local neighborhood is defined to be adjacent sampling positions with similar standard deviation values. An example of this is given in Figure 4.6 where the final trough is denoted by ■. The output of this step is the set of detected crop row lines **L**.

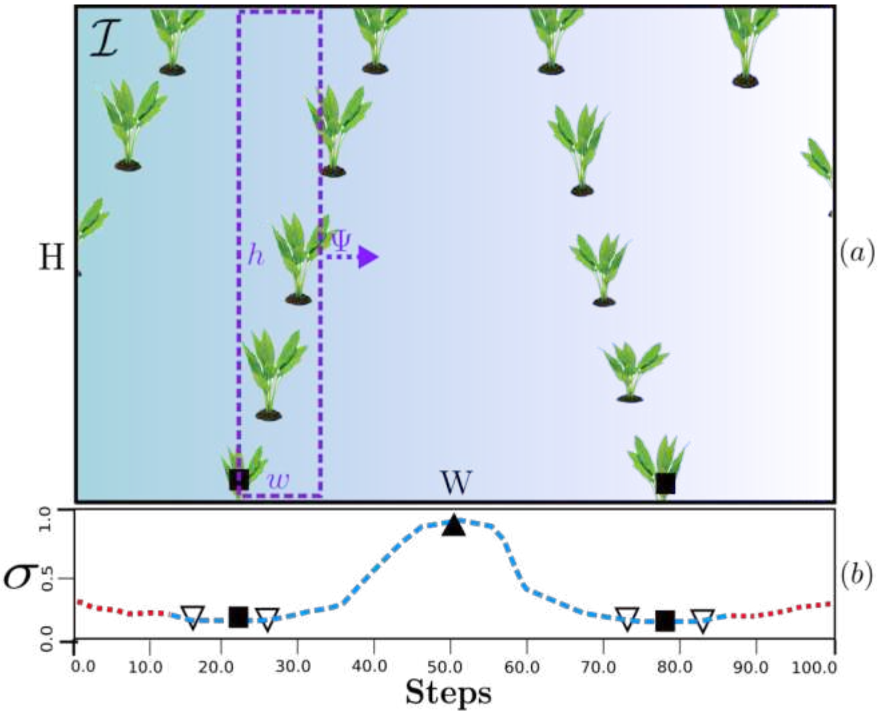


Figure 4.6: The sliding window Ψ is applied progressively. The moving variance of the estimated line angles is used to represent the field structure. The peaks (▲) and troughs (▽) from the field structure are used to find the crop rows and the center of the between crop rows respectively. The weighted average of multiple troughs leads to the final trough ■.

4.4 Visual-Servoing Based Crop-Row Following

To guide the robot within the crop rows, we utilize our previously developed in-row guidance approach [7] which relies on the locally extracted features from camera images used on the image-based visual servoing (IBVS) controller [40, 32]. By continuously regulating the robot's motion based on this controller and observation features as explained in Section 4.3, we ensure the robot stays in the middle of the path and follows the desired crop rows to their end.

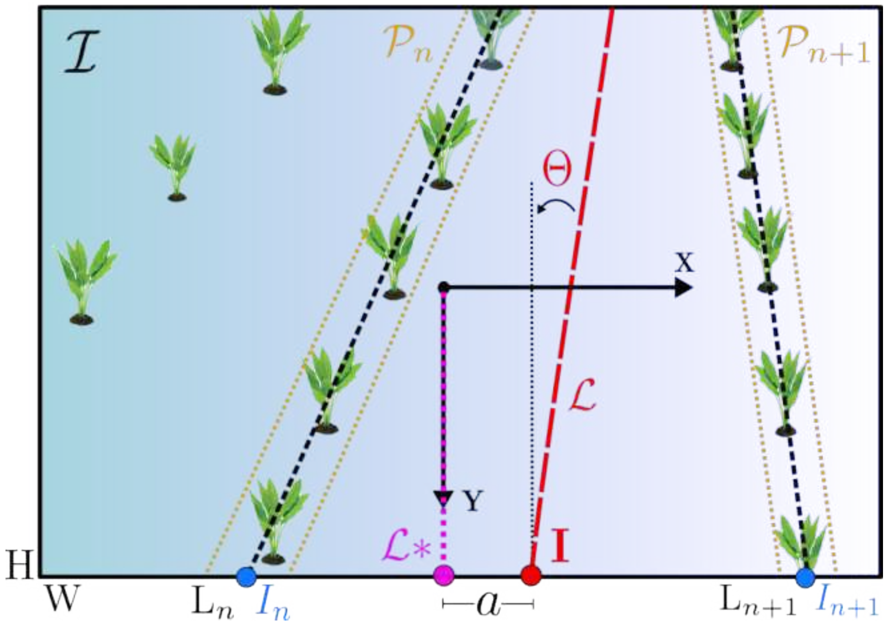


Figure 4.7: The image frame $\mathcal{I}$ and $\mathcal{L} = [\mathbf{I}, \theta]$ denotes the average line estimated from the visible crop rows.

We define the primary guiding feature $\mathcal{L}(t) = [\mathbf{I}, \Theta]$ as an average line derived from previously detected crop row lines in $\mathbf{L}$. In this context, the error function $e(t)$ for the IBVS controller is defined as:

$$e(t) = \mathcal{L}(t) - \mathcal{L}^*, \tag{4.3}$$

The $\mathcal{L}(t)$ line is computed from the current camera image, and $\mathcal{L}^* = [0, \frac{H}{2}, 0]$ corresponds to the desired feature value located at the bottom center of the image $\mathcal{I}$, as depicted in Figure 4.7. The camera's velocity $u_c = (v_c, \omega_c)$ includes both the instantaneous transitional velocity v_c and the angular velocity ω_c of the camera frame. Consequently, the robot's frame velocities u can be expressed as a function of the camera velocity u_c using the relationship $u = {}^{R}T_C\, u_c$, where ${}^{R}T_C$ is a homogeneous rigid-body transformation from $\mathcal{F}_C$ to $\mathcal{F}_R$.

$$\dot{\mathcal{L}} = I_s\, u_c = I_s\, {}^{C}T_R\, u, \tag{4.4}$$

The interaction matrix I_s defines the relationship between the feature dynamics $\dot{\mathcal{L}}$ and the robot velocity in the camera frame u_c. For crop monitoring or weeding tasks, where the robot typically operates at a constant linear velocity $v = v_x^*$, the feature dynamics can be reformulated in terms of the robot's velocity u. The Jacobian matrix J then captures the relationship between these variables, as shown:

$$\dot{\mathcal{L}} = J\, u = J_v\, v^* + J_\omega\, \omega, \tag{4.5}$$

where J_v and J_ω are the columns of the Jacobian matrix J. Therefore, the control law can be obtained by:

$$\omega = -J_\omega^+ \left(\lambda e + J_v v^*\right), \quad \lambda > 0, \tag{4.6}$$

where J_ω^+ indicates the Moore-Penrose pseudo-inverse of J_ω.

Hence, the controls law Equation (4.6) leads the current feature $\mathcal{L}(t) = [a, \frac{H}{2}, \Theta]$ to towards $\mathcal{L}^*$, where H denotes the height of the image and a is the deviation from the image center of the intersection point I. By continuously regulating the robot's motion based on the visual-servoing controller and observation features, we ensure the robot stays in the middle of the path and follows the desired crop rows to their end. At the end of crop rows, we switch to the next lane by using the switching technique described in the next section.

4.5 Multi-Crop-Row Switching

To autonomously navigate over an entire row-crop field a robot must be able to both navigate down lanes and shift between them. Utilizing an only image-based motion controller in conjunction with other localization techniques (like GPS and wheel odometry) could considerably improve the reliability of the system in cases of outage of motion information due to hardware problems and environmental situations. In our previous work [7] the task of changing lanes was managed successfully, however, it was only designed to handle a single crop row in a lane under a highly engineered condition. This method was highly reliant on the seeding pattern of the crops-rows and struggled with cases that often occur in real fields like uneven-seeded crop rows, unexpected distances between the rows, and the appearance of cluttered and weedy regions.

Furthermore, it required significant space to perform the switching maneuver could not reconcile differences between rows, and subsequently followed the incorrect lane. In this work, we propose a multi-crop row switching technique that detects and counts the rows as it progressively shift between them. This takes advantage of the side-ways movement of BonnBot-I allowing easier transitions while requiring less space. To detect a new crop row we exploit SIFT features to ensure we only traverse the desired number of crop rows to confirm our new lane is in the correct location without relying on any motion information neither odometry nor GPS.

To detect a new crop row, we use a feature-based crop row recognition technique based on SIFT features described below, and the number of crop rows to shift by is given by the previous step which gives us the number of crop rows beneath the robot. This approach is designed for arable farms land with multiple rows complying with European phenotypic farm patterns [5]. The process of changing lanes needs to be robust to different numbers of crop rows contained in a lane and ensure complete coverage is obtained without missing or double counting crop rows. The approach described here takes advantage of the side-ways movement of BonnBot-I allowing easier transitions while requiring less space.

We use a feature-based crop row recognition technique based on SIFT features described below. The feature-based crop row recognition technique also reduced issues with lane detection due to its robustness to both false detections and missed detections of the crop rows themselves. We start by considering the robot to have found the end of the crop row; stage ③ of the navigation scheme Section 4.2. The multi-crop row detection algorithm Section 4.3 provides us with the number of crop rows C that have been traversed. To find new crop rows and lane switching, we need to move across C rows and then restart the crop row following the algorithm.

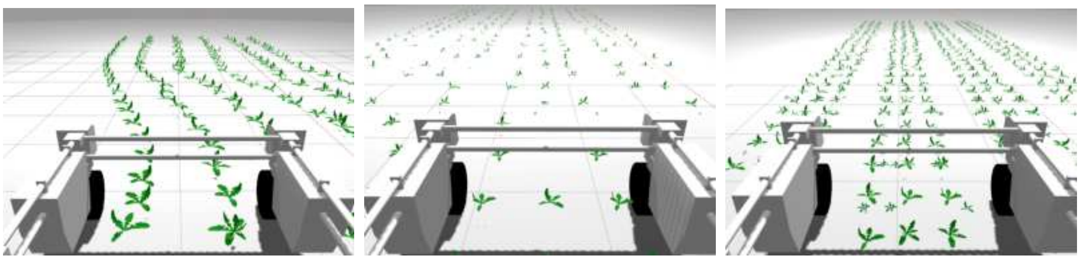
To do this we describe each crop row by a set of SIFT features and follow the algorithm described below. Assuming we are moving left-to-right, we store the features of the right most parallelogram in the image forming a feature set $\mathcal{G}$. The robot then starts moving to the right side with a constant $u_y = 0.15\, m/s$ velocity. Upon receiving a new image we detect the crop rows in a similar manner outlined in Section 4.3 and then only consider the right-most side of the image. We extract a new set of SIFT features from the right-most parallelogram in the image forming a feature set $\mathcal{G}^*$. The new feature set, $\mathcal{G}^*$, is potentially a new crop row. To determine if $\mathcal{G}^*$ is a new crop row we compare it to the stored SIFT features $\mathcal{G}$. If a new crop row has been detected, we update the stored features ($\mathcal{G}$=$\mathcal{G}^*$) and continue this process until we have moved across C new crop rows.

To compare the SIFT features, we use a FLANN-matcher to obtain the best matches between the two sets $\mathcal{G}$ and $\mathcal{G}^*$. This results in the Euclidean distances between $\mathcal{G}$ and $\mathcal{G}^*$ being stored in Ω. We then take the average of the m matches in Ω which are above a threshold λ. This is used to provide a distance measure between the two sets of features: When $D(\mathcal{G}, \mathcal{G}^*)$ exceeds a threshold τ we assume a new crop row has been found, τ is a crop type specific constant.

$$D(\mathcal{G}, \mathcal{G}^*) = \frac{1}{m} \sum_{i=1}^{m} m \tag{4.7}$$

4.6 Experiments

We performed three experiments to show the capability and robustness of our proposed approaches. These experiments were carried out on both simulated and real phenotyping fields. The simulated fields are built in the Gazebo environment with either two or three rows in a lane. It is also designed with various challenging arrangements, such as curved crop rows (Sim-Curved), crop rows with large inter-



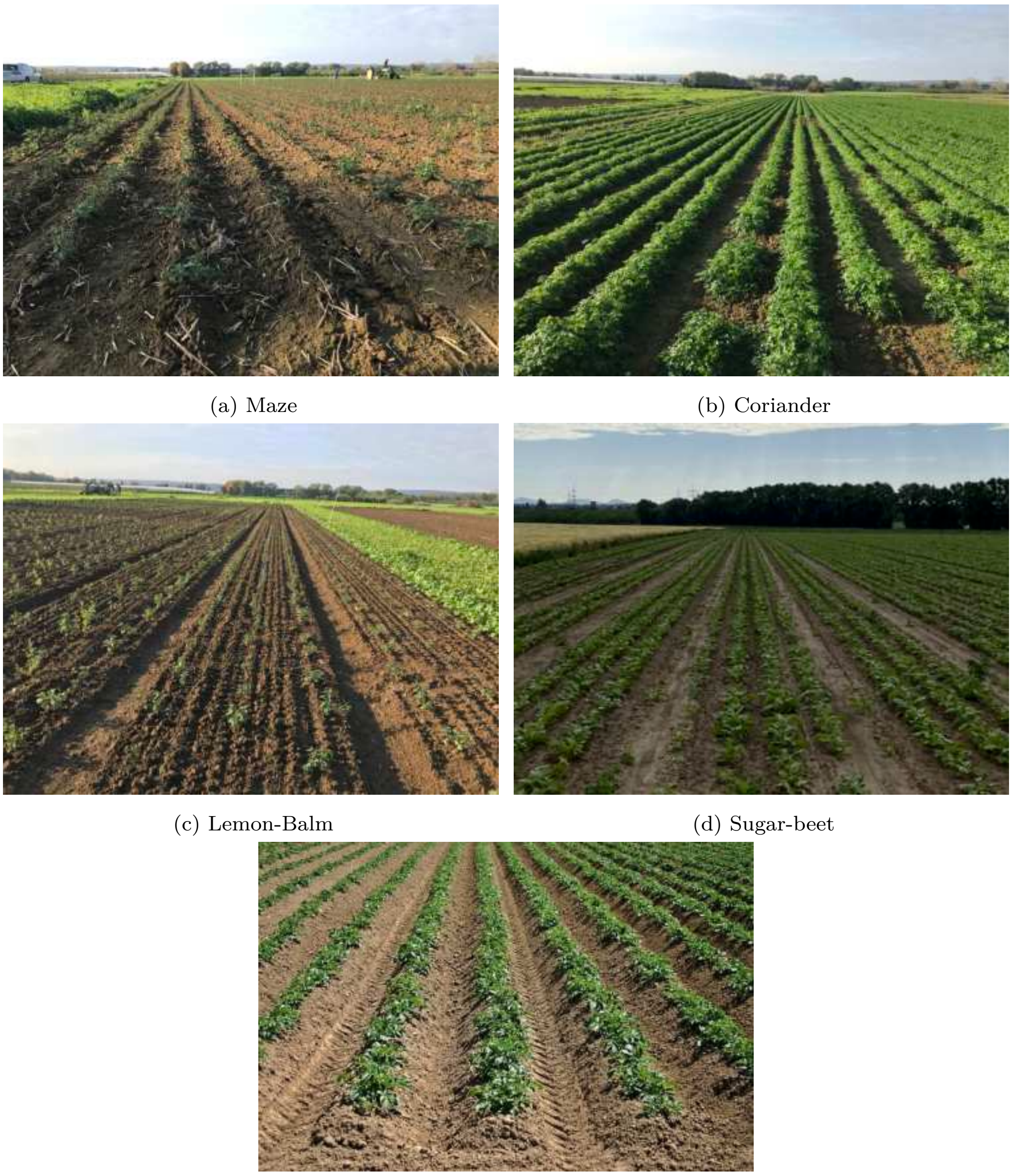
(a) Sim-Curved (b) Sim-Large-Gaps (c) Sim-Dense-Weed

Figure 4.8: Simulated Fields with different plant sizes as weeds (small) and crops (big), for visualization soil background in simulation is removed.

crop gaps (Sim-Large-Gaps), and lanes of crops having dense weed appearance (Sim-Dense-Weed), which are depicted in Figure 4.8. The real fields represent up to five different crops with non-similar canopy shapes and varying numbers of crop rows per lane. All results outlined in this section are based on the evaluation data and require limited human intervention (apart from minor hyper-parameter tuning) during run-time and navigation.

4.6.1 Experimental Setup

The experiments of all five crop types were completed on BonnBot-I at Campus Klein-Altendorf of the University of Bonn under various illumination and weather conditions such as: relatively wet and very dry grounds, cloudy and sunny day times with long and short shadow cases (with minimal hyper-parameter tuning), depicted in Figure 4.9. We used a 1:1 scale Gazebo simulation model with a realistic field to fast-track the field experiments. From this simulation, we were able to derive our algorithmic hyperparameters. On BonnBot-I, the front and back navigation cameras are fixed at a height of $1.0\,m$ and tilt angle $\rho = 65°$. Both camera resolutions are 1280×720 with a capture rate of $15\,fps$. For all experiments, the width w of the sliding window Ψ was kept constant $w = W/10 = 128$ with a height of $h = 720$ pixels. This window size and stride equal to 13 ensures $\simeq 95\%$ overlap between consecutive sliding windows. Also, we empirically set $k = 10$ in Equation (4.2), which in the simulation provided the best trade-off between sample consistency and neighborhood relationship. As the primary goal of this platform is to perform weeding, we set its velocity to be a constant $v_x^* = 0.5\,m/s$. We use differential velocity control within the crop rows and omni-directional control for switching between the lanes. Our approach is implemented using Python and PyCuda, ensuring real-time operation (while with CPU-only machines, performance is real-time) and runs on a user-grade computer (Cincose DS-1202).



(a) Maze

(b) Coriander

(c) Lemon-Balm

(d) Sugar-beet

(e) Potato

Figure 4.9: Long-view of Row-crop fields; the robot was deployed in a variety of row-crop field scenarios characterized by significant variability in illumination, crop type, canopy architecture, and phenological stage.

4.6.2 Multi-Crop-Row Detection

The first experiment is a qualitative analysis of the ability to detect crop rows in the field using the technique described in Section 4.3. We use ExG with Otsu's technique [84] to differentiate foreground from background as this obviates the need for individual hand-tuned plant segmentation thresholds. The goal of this technique is to exploit the dominant crop locations and accurately detect the best location for traversing a lane (i.e., keeping the crop rows under the platform). Due to weeds growing between the crop rows, this can be a challenging proposition

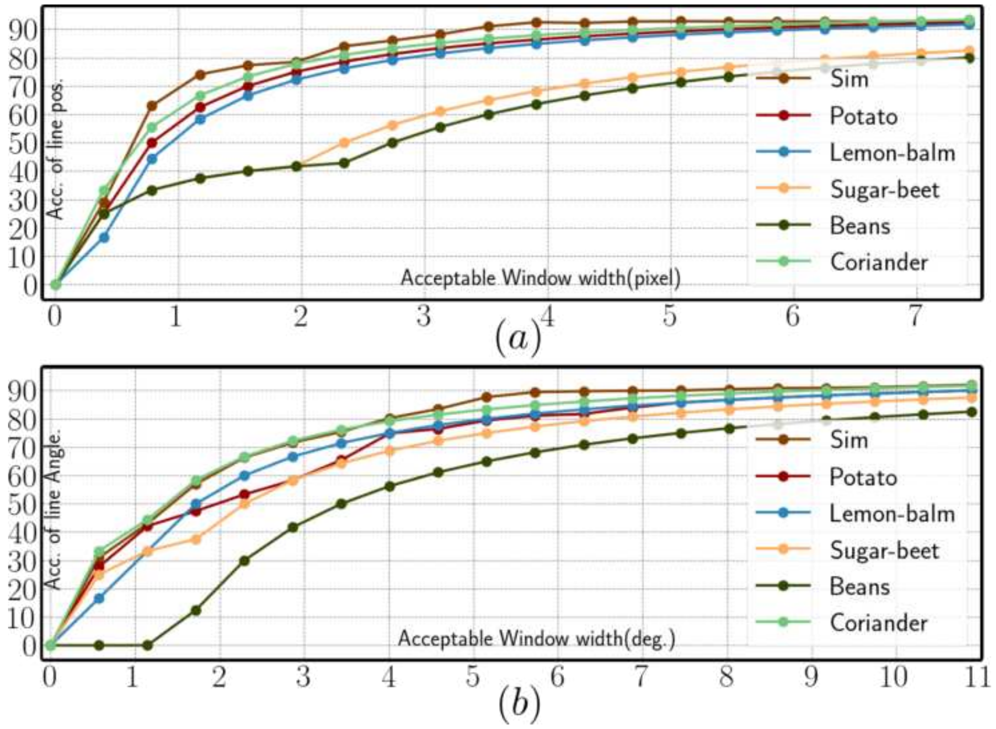


Figure 4.10: Performance of multi-crop row detection technique: accuracy of detections (a) position (w.r.t the image width) and (b) orientation w.r.t the acceptable thresholds

in real fields. Qualitative results for four crops (sugar-beet, coriander, potato, and beans) are presented in Figure 4.12; we refrained from adding Lemon-balm and simulated field results due to space limitations. The illustrated crops have diverse canopy types (see Figure 4.12) and are arranged in two standard patterns with two and three crop rows per lane. In the bottom row of Figure 4.12 it can be seen that our approach for detecting crop row lines through peaks and troughs works for these chosen crops even with a varying number of crop rows. This is true even for challenging crops such as coriander and beans.

For all crops, we were able to consistently detect both the peak (no crop row) and trough (crop row) locations regardless of the presence of weeds. This is especially evident in coriander, where even with the small distance within the crop row (between the coriander plants), we are still able to detect the crop rows. This is an example where dividinga single large region into sub-regions is essential. Sugar-beet, Figure 4.12-a, is another interesting use case. Visually, it is considerably more difficult to discern the crop locations, however, this technique was still able to extract crop rows. Overall, this technique for crop row detection successfully located the required troughs in order to navigate a lane, providing accurate information required for the other stages of our system. To further analyze the robustness of the crop row detection technique, we perform a quantitative evaluate. For each of the five crops (those listed in Figure 4.12 and lemon-balm) and the simulated field, 100 images were annotated using data from

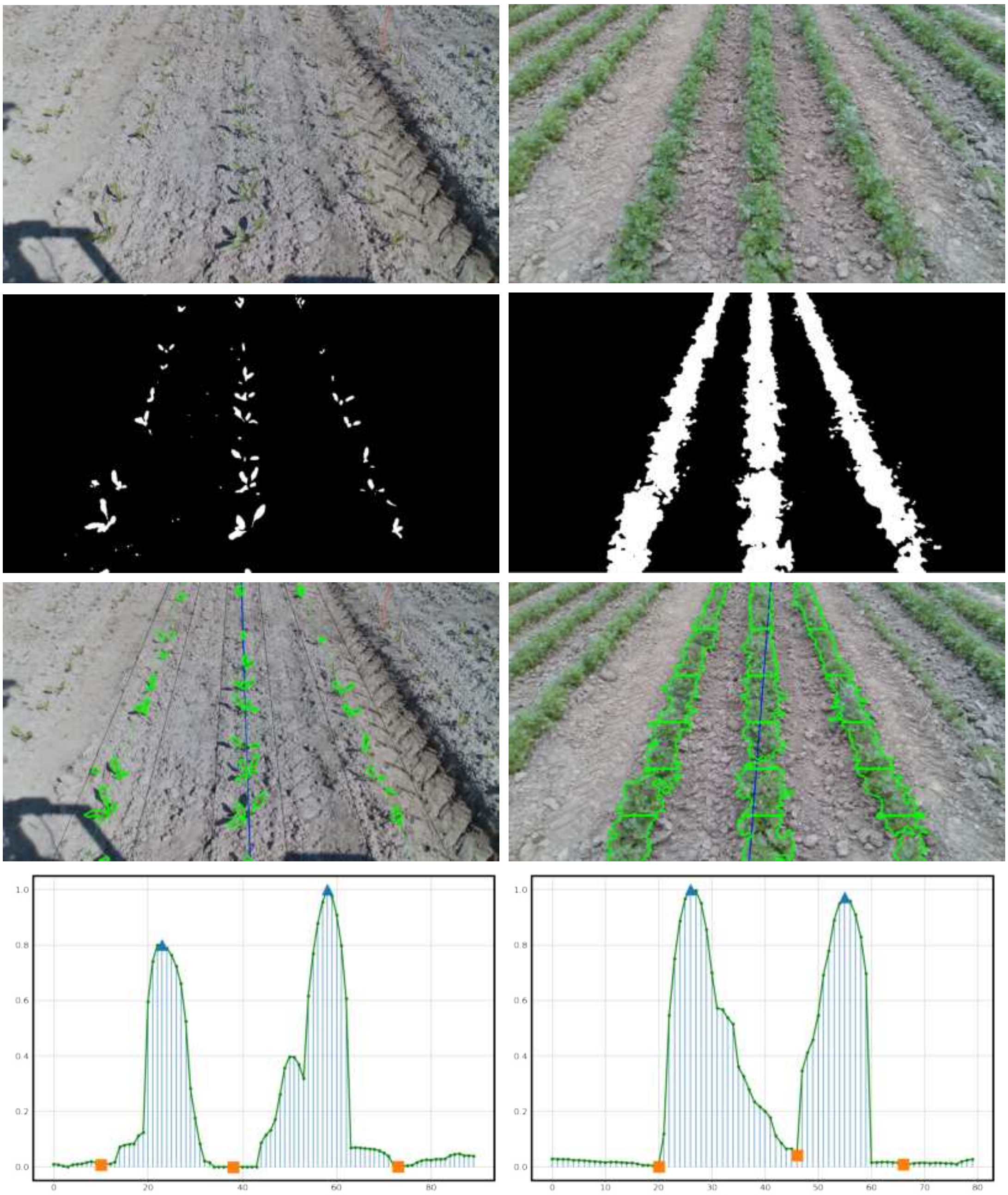
BonnBot-I where the camera tilt angle ρ was varied from 55° to 75°.

The annotations contain the ground truth of all the crop row lines located underneath the robot and crop masks belonging to the main crop rows. To measure the accuracy, we compare the predicted lines of each image to the ground-truth using two parameters: position and orientation. The position of a line is defined based on its intersection with the bottom edge of the image, where the distance between the prediction and the ground truth is normalized based on the width of the image. Figure 4.10 outlines the quantitative performance result of real fields and the averaged performance of simulated fields.

We observe our method is able to estimate crop row positions with a mean accuracy of 88.1% and a standard deviation of 8% over all types of real crops. Similarly, we see that the algorithm is able to correctly estimate the orientation of crop rows in more than 88.3% of cases when the acceptance threshold is set to 11 degrees. The crop row lines of beans and sugar beet were the hardest to estimate. For sugar-beet, we attribute this to the fact that the crop was at an early growth stage, as seen in Figure 4.12, and this made it more complicated to detect the crop lines. For beans, we attribute this to their branchy canopy shape, disarranged seeding pattern, and plant vibration due to the wind in the field. One potential use case where the approach may fail is when the number of weeds is close to or greater than the number of crops in the image (very high weed pressure). Furthermore, difficulties may be faced when navigating the field once full canopy closure has been achieved and there are no visible crop lanes to follow (full vegetation). Nevertheless, overall, we observe that our novel crop row detection method could estimate lines of crops in a variety of challenging real-world conditions for different crop types reliably.

4.6.3 Navigating Along The Crop-Rows

To analyze the performance of our crop row navigation technique, we require accurate ground truth information. We collected the ground truth information by manually driving the robot down each of the row-crop fields for all crop types and stored the associated information (e.g., GPS measurements) for later evaluation. Also, all simulated crop rows came with reference line coordinates from the simulation environment. The associated GPS measurements are then used as the "correct" position (accurate to $1cm$). Even though manual operation can cause some errors, we consider this to be an appropriate ground truth to compare t,o as the crop rows are not guaranteed to be planted in a straight line. The five crops (sugar-beet, coriander, potato, beans, and lemon-balm) provide a range of challenges such as different canopy types, weed densities, and varying growth stages. Table 4.1 outlines the performance of our full pipeline, including the navigation system, on these crops as well as three challenging simulated fields. The

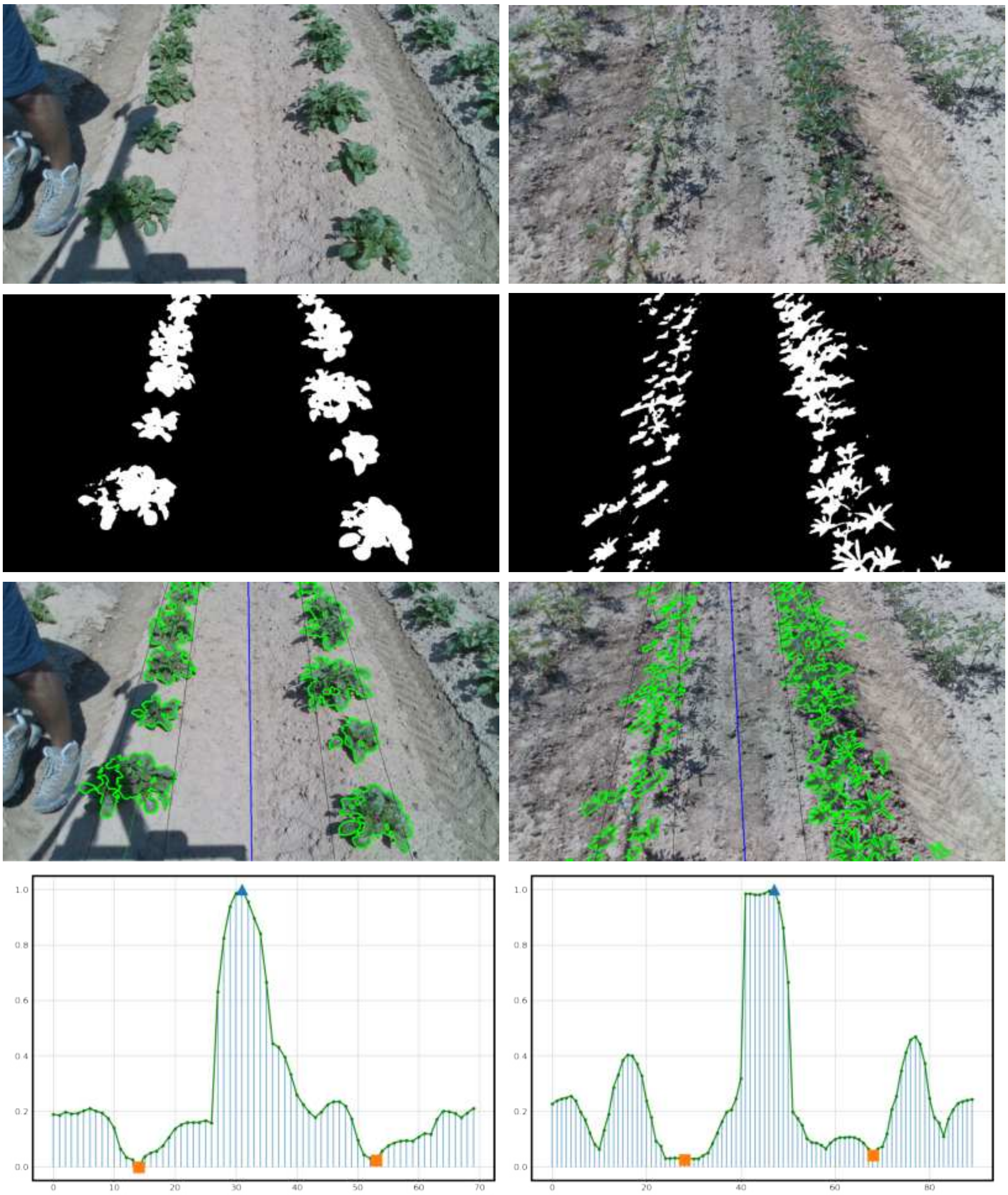


(a) Sugar-beet

(b) Coriander

Figure 4.11: Illustrations of four crops (top row) RGB images with marked vegetation index of detected rows and their corresponding field structure signal (bottom row). The detected peaks (blue triangle) and troughs (orange square) obtained via their prominence in the signal are also provided. Denote that the field structure signal only includes the values of lines intersecting with the bottom axes of the image.

most challenging crop for navigation was sugar-beet, and we attribute this to two reasons. First, the crop was at an early growth stage, as seen in Figure 4.12, and this made it more complicated to detect the crop lines. Second, not all of the sugar-beet had germinated and this led to gaps or long "dead space" along the



(a) Potato

(b) Beans

Figure 4.12: Illustrations of four crops (top row) RGB images with marked vegetation index of detected rows and their corresponding field structure signal (bottom row). The detected peaks (blue triangle) and troughs (orange square) obtained via their prominence in the signal are also provided. Denote that the field structure signal only includes the values of lines intersecting with the bottom axes of the image.

rows, which the same effect can be seen in Sim-Large-Gaps results too. However, tracking multiple crop rows allowed our technique to still navigate over the entire evaluation area without any manual intervention. This evaluation shows that multiple crop rows following have considerable benefits over techniques that only

Table 4.1: Lane following performance of BonnBot-I using the proposed method in real and simulated fields.

Crop	**Length**	$\mu \pm \sigma$ **of dist. to crop rows**	$\mu \pm \sigma$ **of angular error**
Sim-Curved	200 m	9.01 ± 2.63 cm	4.52 ± 3.52 deg
Sim-Large-Gaps	200 m	6.75 ± 3.15 cm	4.76 ± 2.69 deg
Sim-Dense-Weed	200 m	7.41 ± 2.86 cm	3.91 ± 1.73 deg
Beans	52 m	3.49 ± 2.89 cm	3.73 ± 3.21 deg
Potato	37 m	2.18 ± 3.01 cm	4.91 ± 1.63 deg
Coriander	54 m	2.91 ± 2.38 cm	2.57 ± 1.05 deg
Sugar-beet	69 m	8.41 ± 3.79 cm	3.25 ± 1.27 deg
Lemon-balm	40 m	2.12 ± 1.58 cm	3.21 ± 2.83 deg

track a single crop row.

From a navigational perspective, the bean crop had a large standard deviation between real fields and Sim-Curved among simulated fields when considering angular error. The weather conditions played a crucial part in this, as heavy winds consistently changed the location of the leaves of the crops. This limitation in the navigation technique leads to large angular variations while traversing the lane.

Across the five real crop types, the average deviation from the ground truth was $3.82cm$ or approximately 10% of the crop row distance. This minor fluctuation is sufficient to ensure safe navigation without damaging crops. Finally, this navigational accuracy was sufficient for the technique to traverse all the crops in the field without manual intervention.

4.6.4 Multi-Crop-Row Switching

Our final evaluation is based on the lane-switching technique outlined in Section 4.5. To evaluate the performance of this technique, we manually annotated randomly selected positive and negative samples from our three main crop types: beans, coriander, and sugar-beet, and simulated fields as we did not have the switching information for potato and lemon-balm due to a technical problem. We store one of the positive annotations as our main row and compare it to each of the other positive and negative rows. Figure 4.13 outlines the precision-recall curves achieved on each of the main crops. We outline the F1 score here which provides a trade-off between precision and recall. For this simple matching technique, we are able to achieve promising results across all crop types, even sugar-beet, which, as outlined Section 4.6, had a number of added complexities. The early germination stage of the crop added extra complications to crop row switching as, even visually, the rows appeared similar. However, we were able to

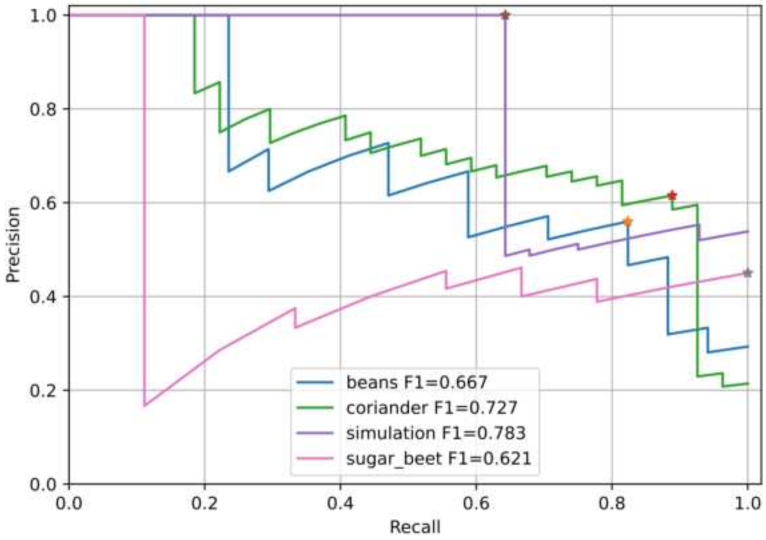


Figure 4.13: The precision-recall plot for switching the platform across lanes, includes beans, coriander, and sugar beet.

achieve an F1 score of 62.1, which was the lowest-performing crop. Overall, from these evaluations, we were able to empirically set thresholds that favored high precision in order to remove false positives. From this, we were able to provide a lane-switching technique that was robust to the challenges of each crop type. In experiments in the field and deployed on a robot, it successfully switched 6 lanes of crop, across the three main crop types, without any manual intervention. Furthermore, we used this technique in simulated crop row fields for 20 lane switching cases, which outlined an average success rate of 90%.

A final key analysis of our crop row switching technique is the distance needed to perform the maneuver. In our experiments, an average of $0.7m$ was required from the end of the crop to the location of the camera. This is a marked improvement over [7], which required more space to perform the switching than the length of the robot itself.

4.7 Conclusion

Autonomous agricultural robots are increasingly important in enhancing farm productivity by performing precise field management tasks. Nowadays, robots and agricultural vehicles navigate using high-precision dual-frequency RTK-GNSS receivers, which guide them along pre-programmed paths. However, the cost and vulnerability of such systems to outages have driven interest in alternative navigation methods.

In this chapter, we introduced an approach exploiting the crop row structure using only the local observation from the onboard cameras without requiring any global or local position awareness. Despite the potential, these systems face significant challenges, such as working in highly self-similar enrolments and unreliable sensor data due to the dynamic nature of crop fields. Furthermore, the changing field environment, driven by plant growth, necessitates frequent updates to field maps or the adaptability of used algorithms. To achieve this, we have proposed a novel multi-crop row detection strategy that can deal with cluttered and weedy scenes.

Furthermore, we proposed a novel lane-switching strategy enabling BonnBot-I to switch to a new lane independent of any global positioning system or human intervention robustly. We evaluated our approach on BonnBot-I on up to five crop types (with varying canopy shapes) in real field conditions and three challenging simulated fields achieving an average navigation accuracy of 3.82cm in real fields.

BonnBot-Is multi-crop row visual navigation system and our previous work [7] have significantly influenced the future of automated plant phenotyping and intervention. To further support the robotics community, we released the Multi-crop row dataset necessary to reproduce our results and open-sourcing the code of our approach. This impact is evident in the number of publications and projects initiated using our implementation. A comprehensive collection of qualitative results, along with the C++ and Python implementations, can be accessed in `https://github.com/PRBonn/visual-croprow-navigation` and `https://github.com/Agricultural-Robotics-Bonn/visual-multi-croprow-navigation` to facilitate further advancements in this field.

Nevertheless, the classical vision-based approaches especially in outdoor navigation applications struggle in dynamic and cluttered environments, particularly in distinguishing between crops and weeds in densely vegetated areas. While these approaches excel in controlled and engineered environments, they are mostly sensitive to varying lighting conditions, occlusions, and the natural variability in crop and weed appearances, where precise differentiation is critical for effective field operations. Hence, future works could explore an alternative approach to detecting individual plants (crop/weed semantic segmentation) using deep neural networks to surpass classical signal processing and machine-vision techniques to move towards a holistic and crop-agnostic navigation approach for row-crop fields.

In the next chapter, we shift focus to the third research question, addressing plant-level targeted intervention using robotics and robotic vision. This includes reviewing precision farming approaches and defining system requirements, exploring novel techniques for target-space management, path-planning strategies, and a biodiversity-aware weeding scheme. This is followed by presenting the results of experiments conducted in both simulated and real-world settings to evaluate their performance.

Chapter 5

In Field Bio-Diversity-Aware Interventions

In this chapter we present our novel precision weeding approach. This approach enables plant-level and bio-diversity-aware interventions in the fields using our self-built robotic platform BonnBot-I (introduced in Chapter 3). In this chapter, we show how we fuse image-based perception and geometric information of the plants underneath the robot to properly guide independently controllable weeding axes of BonnBot-I in real-time while the robot moves. This

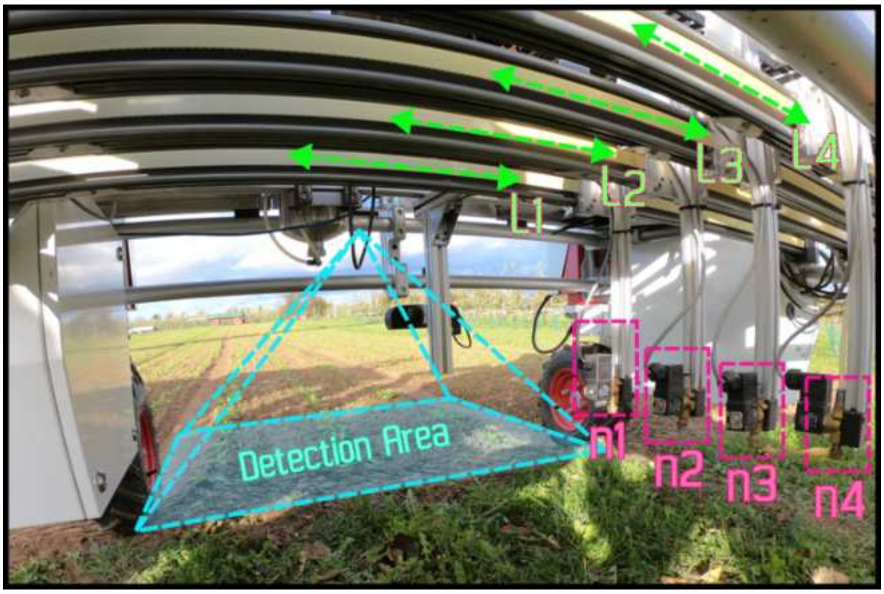


Figure 5.1: BonnBot-I Platform, a robotic platform capable of conducting field monitoring and precision weed management in arable field. The detection camera positioned at the front of BonnBot-I observes the wheel-to-wheel area beneath the robot (depicted as a cyan rectangle). Linear axes of the weeding tool (L1-L4, shown as green arrows) in the back of the robot carry spray nozzles (N1-N4, indicated in pink).

work was originally introduced in [5]. Furthermore, in [6] for the first time we introduce the concept of plant-level bio-diversity-aware weeding which is integrated into the BonnBot-I weeding approach.

The preference for consuming more natural and organic foods has rapidly increased in recent years [23]. This has forced the agricultural industry to use fewer agri-chemicals when dealing with weeds while maintaining the quality and quantity of their crops. Weeding is an important aspect of arable farming as weeds compete for nutrients in the soil [97], potentially reducing yield. To alleviate the impact of weeds, unifo9rm treatment has been the most common approach. This means that regardless of the presence of weeds the entire field is sprayed with herbicide. This approach has led to an increasing number of herbicide-resistant weed species [127] as well as negatively impacting the environment by increasing soil erosion and water contamination [74].

In Chapter 3, we introduced the BonnBot-I platform and outlined the design of its sensory setup and its novel intervention tool. However, a key component of BonnBot-I is its ability to perform precise intervention actions in the field. These actions are derived from the ability to use advanced planning and fusion algorithms which are built into the platform.

In [5], we initially developed a planning approach for BonnBot-I that used non-overlapping segments of field observation to predict intervention trajectories for weeding axes of the weeding tool. In a follow-up work [6] we proposed an alternative rolling-view planning approach that improves the intervention accuracy and performance. Furthermore, it emphasizes the need for robots to recognize and adapt to the diversity of weed species. By integrating such details in the intervention scheme, we could equip the BonnBot-I to perform bio-diversity-aware weed management. This advancement in targeted plant-level intervention enhances crop protection and resource optimization in weeding, signifying a shift towards more nature-conscious and efficient agricultural practices.

Overall, in this chapter, our third research question is "*How can we enable a weeding robot to perform crop-agnostic, plant-level weeding while considering biodiversity-aware practices?*" of this thesis by offering the following main contributions:

- Propose **several novel concepts for per-plant intervention approaches** towards enabling more sustainable weed management practices.

- For the first time we propose the integration of **bio-diversity aware weeding** in the capabilities of an autonomous robotic weeding platform.

- We introduce **a framework for testing different weeding strategies** using a simulation environment.

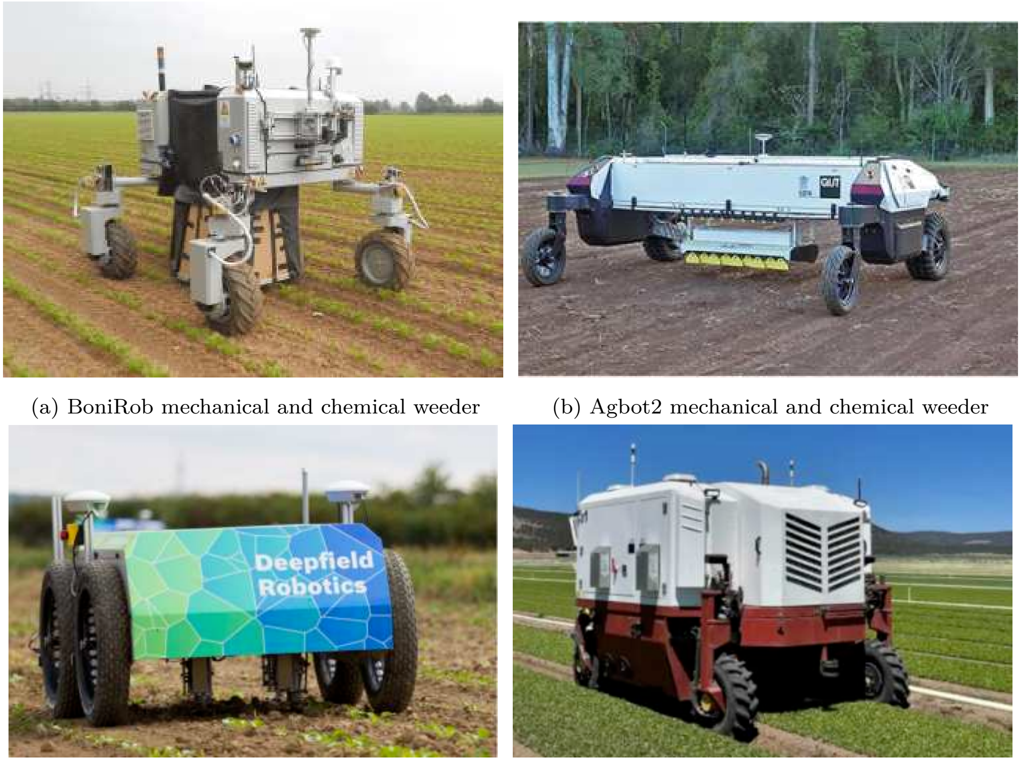
- We represent the results of **real-world experiments** of multi-nozzle weeding of BonnBot-I.

This chapter is structured as follows: we begin by reviewing similar robotic weeding approaches addressing precision farming. In Section 5.2, we define a standard weeding scenario and outline the system requirements for interventions. Section 5.3 introduces our methodology for transforming real-field observations into a deployable space for weeding. Next, in Section 5.4, we discuss target-space management for allocating targets across multiple weeding axes in real-time. In Section 5.5, we present path-planning strategies for controlling BonnBot-I's intervention heads, including a biodiversity-aware weeding scheme in Section 5.6. Finally, Section 5.7 describes experiments in simulated and real-world scenarios, evaluating the systems performance in sugar beet and corn fields. The chapter concludes with a discussion of results and potential improvements.

5.1 Related Works

In recent years, weed management techniques have rapidly evolved to treat individual weeds precisely. Intervention on this level cuts the amount of agrochemicals required which minimizes the environmental side-effects. Robotic weed intervention has the potential to revolutionize weeding paradigms through plant-level weed management. To achieve this, the treatment of each plant must be dictated by its species, size, and impact upon not only the crop but also the environment [22, 52]. Several robotic weed control platforms have been introduced offering active and passive interventions. On these platforms, a variety of weeding implements have been investigated including physical [27, 72, 15, 9], chemical [119, 43], electrocuting [10], laser-based [107, 87, 107, 121], and multi-intervention system [90]. Considering the need to operate in different fields with varying crops, weed species, and weed distributions, it is clear that there is no one best general solution and robotic systems should be able to cater to a variety of tools [72, 6].

Slaughter *et al.* [98] provides an early detailed review of different tools, While not specifically researching a platform. An early, low-cost, platform was developed by [81] which contained a mechanical weeding tool for intra-row intervention that required a human in the loop. Bawden *et al.* [19] proposed an automated platform that utilized a row of weeding hoes and spray nozzles to improve broadcast applications and perform multi-modal weeding (physical and chemical). Also combining mechanical tools (two ranks of stampers) and sprayers [80] can selectively weed based on the overall size of the plant. To design a spraying platform [123] developed a technique that was able to specifically target regions or



(a) BoniRob mechanical and chemical weeder

(b) Agbot2 mechanical and chemical weeder

(c) Deep field mechanical weeder

(d) Carbon robotics laser weeder

Figure 5.2: Weeding Platforms; each of these platforms are designed with specific weeding tools and operates in unique ways. (a) BoniRob [88] , (b) Agbot2 [73] , (c) Deepfield Robotics [86] , and (d) Carbon robotics [87]

weeds. In an indoor experiment, they were able to reduce the amount of chemicals used, compared to a uniform sprayer, by 46.8%. Chang *et al.* [28] built a platform to evaluate two different mechanical weeding tools which was tested on a purpose-built field with a single crop-row of 20m. Their approach used Yolo to both locate weeds of interest in the field but also control the movement of the platform.

A consistent trend with the above approaches is that they aim to improve broadcast or uniform weeding by introducing equally spaced tools. In doing so they ensure coverage but have the significant downside of having to replicate the tool across the entire width of the robot. To achieve a truly precise weed control method, a weeding platform must be equipped with the capability to carry and use multiple tools simultaneously. Replicating equally spaced tools across multiple sets increases the complexity and manufacturing challenges of this task.

A more promising approach involves using movable tools on a robotic platform, but this necessitates a sophisticated planning algorithm to ensure efficient and accurate operation. A frequently overlooked aspect of weeding is planning field-based intervention. If a robot carries multiple movable tools then planning their deployment is essential, yet limited work has explored this aspect.

Lee *et al.* [61] is one of the few works in this area. They presented a multi-query approach for efficiently planning paths using a single UR5 robot manipulator to enable precision weeding. This was achieved by maintaining a database of pre-computed paths that were constructed offline and the optimal path was chosen and adapted online. Xion *et al.* [121] proposed a laser-based intervention scheme that segmented the visible weeds into equally spaced regions along the x-axis, this then allowed them to engage targets sequentially.

In this work, we aim to address the limitations of existing weeding platforms by introducing a novel approach. We equipped BonnBot-I with a novel weeding tool (introduced in Section 3.1.4), where a set of movable and replicated tools are utilized to conduct selective operations in real fields. This innovative design offers great flexibility and precision in weed control, enabling targeted intervention without compromising efficiency.

5.2 The System Overview

BonnBot-I is equipped with a novel weeding tool design enabling high-precision plant-level field interventions. It consists of a set of replicated linear actuators and is controlled via the intervention controller unit consisting of several components which are elaborated in the following sections. Here, we briefly explain the conceptual design of the weeding tool, its requirements, and operation assumptions. The most critical challenge in this design is to efficiently plan paths for intervention heads, such that we maximize the number of well-managed weeds in a weeding scenario. A typical weeding scenario is composed of the following stages:

1. Weeds get detected, classified, and tracked within the viewable area of the down-facing detection camera in front of the robot. The monitoring system introduced in Section 3.4 runs the Mask-RCNN network for instance-based semantic segmentation and classification is used to estimate necessary phenotypic information about the plants. Finally, the plants are tracked (tracklets) through the viewable area.

2. The tracklets, which are identified as valid plants for management, are monitored beneath the robot using its localization system. These tracklets are then relayed to a target-space manager, who efficiently allocates them among the various weeding axes for optimal handling.

3. Each intervention planner plans an optimal path considering phenotypic factors like weed type, size, harmfulness, required action time, and biodiversity considerations.

4. The computed paths are transmitted to the relevant axis drivers and nozzle controllers to carry out corresponding actions on the plants.

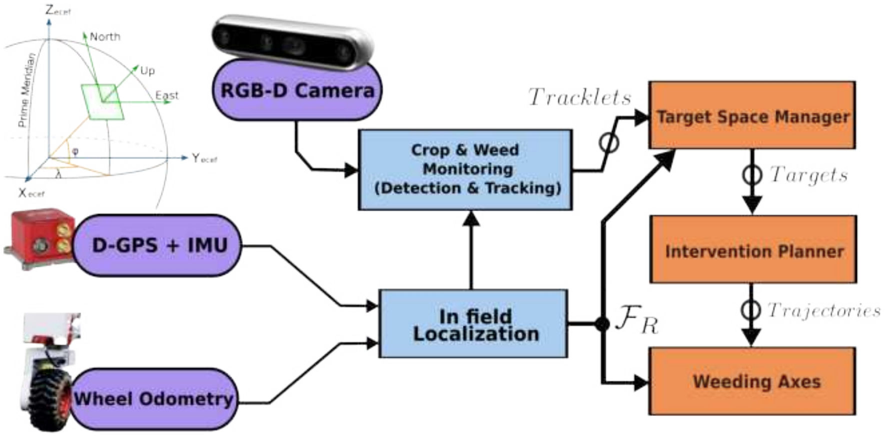


Figure 5.3: The software architecture, including sensors (purple), vision perception, localization of robot base frame $\mathcal{F}_R$ (light blue), intervention planning, intervention planner, and weeding axes controllers (orange).

5.3 Observation Model

In [5] we introduce a segment-view observation model which was used to transfer real-time detection of monitoring systems to comprehensive model field models. Then this model was used to manage planning observations for weeding scenarios at each timestamp. This method used discrete non-overlapping segments of observation to represent a weeding scenario in any time. This segment-view-based model allows us to focus on an individual field segment, enabling us to cope with their distinct weed distributions. We derive the distance between the weeding equipment and individual weeds assuming a weed distribution yielded by a Poisson process with an arrival rate of $\eta = \lambda \times \Pi$, where Π denotes the weeding width illustrated in Figure 5.4. To understand the spacing between the weeds (δ_x) we use the robot's motion along the x-axis within its frame of reference ($\mathcal{F}_R$) depicted in Figure 5.4. While this method was successful, it is less adaptable to the changing weed spatial distribution patterns and thus harms overall performance. To overcome this we propose a rolling-view observation model.

We propose a rolling-view model which, integrates discrete camera observations into a comprehensive model of the entire field. Expanding the planning scope beyond single segments, allowing for more effective weeding strategies over a larger area. A key advantage of this approach is illustrated in Figure 5.4 which highlights that the previous segment-based approach would treat two views independently whereas the rolling window approach updates the planned intervention

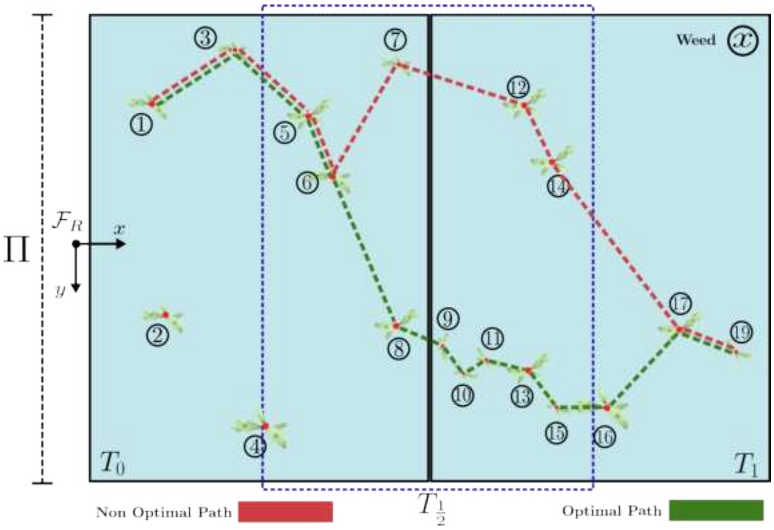


Figure 5.4: Segment-view vs Rolling-view Planning; two separate segments T_0 and T_1 with different weeding distributions are shown. An intermediate T_i is substantially helping the planner to optimize the planned route of the weeding axis (green) w.r.t the baseline (red).

across these two views iteratively. This enables the planning to be optimized more flexibly without increasing the planning window size. By simply incorporating only one intermediate frame $T_{\frac{1}{2}}$, we achieve a notable enhancement in the weeding trajectory and overall performance.

We have implemented this algorithm using multi-threading and dynamic programming techniques that enable the rolling-view planning to be at a frequency > 500Hz on CPU (Intelő Core i7-12700K). The rolling window planner uses the most recent information. This means we take the prior plan and update it only if there are new plants that need to be treated. This rolling-view model not only refines the weeding process and inter-image tracking but also compensates for the vision system's shortcomings, like missed detections or incorrect classifications. The accumulation of multiple detections over the same area considerably increases the accuracy of our predictions. Furthermore, by analyzing continuous field segments, we can predict more precise weeding paths, leading to improved overall efficiency in weed management.

5.4 Target-Space Management

When a robot encounters multiple targets $\mathcal{N}$ beneath it, the primary objective is to engage each target using one of its intervention heads $\mathcal{H}$ while the robot is in motion. Hence, before these targets enter the workspace of the weeding tool, it is crucial to plan the motion for each intervention head. Considering sets of weeds in any given workspace motivates a multi-query approach.

In the proposed workflow, the intervention controller node receives the de-

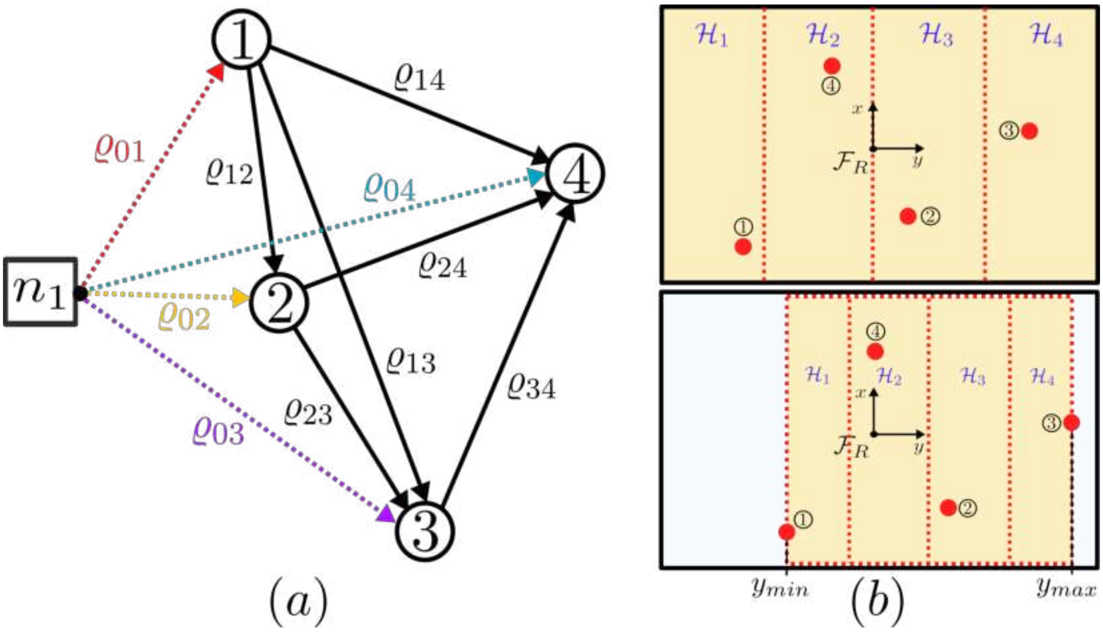


Figure 5.5: In (a) an example unidirectional constrained node-graph is presented where the numbered circles represent weeds (targets) for one intervention tool n_1 with different possible start paths (in different colors). In (b) we present a visualization of static work-space subdivision (top) and dynamic work-space sub-divisions (bottom).

tected targets at time $t + \tau_d$ where τ_d is the time required for detection in the monitoring node. The monitoring node provides plant-specific information like plant category, pixel-wise segmentation, estimated area, and the bounding box. Furthermore, we estimate plant centers based on their predicted segmentation center of mass. This information is then used in the target-space management step to assign targets to the intervention heads. The next step find the best motion plan for each intervention head by maximizing the number of targets that are visited (sprayed).

The main goal is to visit all the targets with at least one of the intervention heads as they pass the workspace of the weeding tool without the need to stop the robot. In our wedding scenario to robot only moves in a forward direction which makes the intervention time-critical. Hence, we use a uni-directional constrained node graph to model the target space.

To obtain the global spatial order of targets in a segment we use the δ_x of each weed (see Figure 5.6). In Figure 5.5(a), each node (circle) shows a weed along with the connecting path between nodes j to k represented with a uni-directional link (arrow) l_{jk}. The link l_{jk} exists if, node j geometrically is located after node k in the 3D world frame $\mathcal{F}_w$. Furthermore, the link l_{jk} is associated with an inter-weed cost ϱ_{jk} based on the distance of nodes j and k and a property denoting motion probability of P_{jk} based on Equation (5.4). We calculate inter-weed costs using the top-right of the cost-matrix $\mathcal{G}_{\mathcal{N}\times\mathcal{N}}$ (to respect the weeds' spatial order).

There are $\mathcal{H}$ independent intervention heads so multiple plans can lead to the same targets being visited (sprayed). To solve this problem, we consider the weeds as a set of targets detected in one location, this motivates us to assign intervention targets to the $\mathcal{H}$ heads as either distance-based or work-space division-based assignments.

1. ***Distance-based Target Assignment (D)***: In this approach, target j gets assigned to the laterally closest intervention head along the sliding direction (y-axis). This means the selected intervention head i has the least motion required to reach the weed j. The lateral distance between heads and weeds is defined based on the 2D Euclidean distance between the projection of the intervention head's position on the ground plane and weed positions on the same plane w.r.t the $\mathcal{F}_w$ frame.

2. ***Static Work-space Division-based Target Assignment (SD)***: In this method, we divide the work space of the weeding tool to $\mathcal{H}$ sub-sections of width $\Pi/\mathcal{H}$ meters. Hence, each intervention head is only responsible for engaging with weeds laying within its sub-work-space as shown in Figure 5.5(b)-top.

3. ***Dynamic Work-space Division-based Target Assignment (DD)***: In this model, for each new set of detected weeds, we first determine the minimum region of intervention defined by y_{min} and y_{max} (see Fig. 5.5 (b)). The minimum region of intervention is then divided into $\mathcal{H}$ equal sub-regions. This process assists in optimizing the planning for weed engagement by potentially reducing the area any one tool has to cover.

5.5 Plant-level Treatment In Field

We assume the robot moves along a crop-row with constant speed γ. Consequently, intervention is time-critical and must respect the spatial ordering of the weeds. There is a constant gap (Γ) between the tools and the area sensed by the camera (C_{detect}). Similar to [18] we assume weeds are uniformly distributed in the field with density λ weeds/m^2. Hence, using a Poisson process we can explain the distance between the weeding implement and individual weeds by accounting for the arrival rate of $\eta = \lambda \times \Pi$. We use the motion along the x-axis of the robot frame $\mathcal{F}_R$ to explain the weeds interval distance (δ_x), visualized in Figure 5.6. This can be shown using the following probability density function,

$$f(\delta_x) = \lambda\Pi e^{-\lambda\Pi\delta_x}, \tag{5.1}$$

also the location of weeds on the y axis can be represented via a uniformly distributed random variable y as,

$$f(y) = \begin{cases} \frac{1}{\Pi} & for \;\; 0 \leq y \leq \Pi \\ 0 & otherwise \end{cases} . \tag{5.2}$$

To engage the i-th intervention head with the j-th weed it has to traverse,

$$\delta_y^{ij} = |h_i - n_j|, \text{ where } 0 \leq \delta_y^{ij} \leq \Pi, \tag{5.3}$$

where h_i is the current position of i-th intervention head and the n_j denotes to the position of the j-th weed.

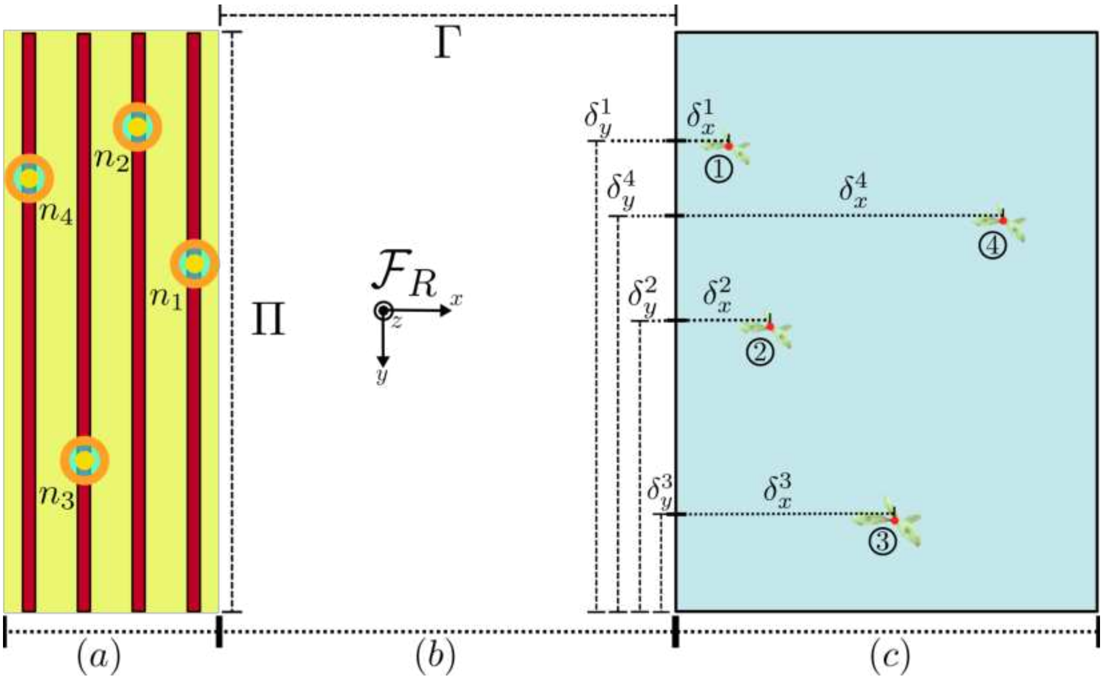


Figure 5.6: Visualization of Kinematic model of weeding tool work-space (a) the weeds detected in the viewable area of the camera C_{detect} (c); and (b) the gap between two regions.

We need to generate $\mathcal{H}$ independent and efficient routes to guide the intervention heads, taking into account assigned targets, the intervention head's position, the robot's linear speed, and the speed and acceleration limits of its linear axes. To do this, the intervention controller embeds all the details about each plant in a uni-directional graph, including its type, segmentation, estimated size, boundaries, plant center, and corresponding bio-diversity characteristics. In this graph, each node represents a detected plant and every edge scores the motion feasibility between two nodes. As introduced in [5], the probability of visiting the j-th weed with i-th weeding nozzle is computed as follows,

$$P_{ij} = P\left(\frac{\gamma}{\vartheta} < \frac{\delta_{ij}^x}{\delta_{ij}^y}\right), \tag{5.4}$$

where ϑ and γ denote the maximum velocity of linear axes and maximum linear velocity of the robot, δ^x and δ^y denote the relative distance between the i-th nozzle to the j-th weed. In this scheme, we define each edge to consist of a feasibility score Γ_{jk} using the logistic function Equation (5.5) of motion between nodes (weeds) j and k.

$$\Gamma_{jk} = \frac{1}{1 + e^{-\omega(S_{jk})}}, \tag{5.5}$$

where S_{jk} is the favorability score in seconds

$$S_{jk} = \frac{\delta^x_{jk}}{\gamma} - \frac{\delta^y_{jk}}{\vartheta}, \tag{5.6}$$

and the weighting parameter ω adjusts how quickly the favorability score makes the Γ_{jk} change from the boundary score (0.5) to being very likely (1.0). The boundary score occurs when $S_{jk} = 0$ and represents when there is just enough time for the tool to transition from node j to k.

In a greedy algorithm, every potential route is calculated by permuting all nodes in the graph, considering the edge directions in the node graph. The best route is the one that visits a high number of nodes with high feasibility. The best performed method utilized in [5] was nOTSP which is a modification of the conventional traveling salesman problem.

Here we address how to plan $\mathcal{H}$ independent efficient routes. The planned routes must guide intervention heads through all their assigned targets while minimizing the chance of missing any target. This has to take into account the the prior knowledge of an intervention head's position, robot linear speed as well as the limits of speed and acceleration of linear axes.

The planning approach generates m potential trajectories $\vec{\mathbf{T}} = [\vec{T}_0, \ldots, \vec{T}_m]$ for each intervention head. Each trajectory $\vec{T}_i$ is an ordered list of length q consisting of weed positions which can be visited. To obtain $\vec{\mathbf{T}}$ we use the following approaches:

1. ***Brute-Force***: In this case we compute all possible routes by finding the permutation of all nodes in the graph (without considering the direction of links). Then the routes with the lowest cost and maximum success rate will be selected from all predicted routes.

2. ***Open Loop Traveling Salesman Planning***: This approach, termed OTSP, is a variant of the classic travelling salesman problem where the agent must visit all nodes of a graph once without making a loop back (Hamilton loop) to the start node [33]. To solve this we use an approach similar to nOTSP where the agent only needs to visit n nodes in the graph, however, in our problem setting we aim to maximize the number of visited nodes while considering other important criteria like cost and success rate.

> We use our constrained uni-directional node-graph representation as a base for solving nOTSP using dynamic programming.

The optimal trajectory for each intervention head is obtained by considering two criteria: the number of nodes successfully visited and the total movement of the predicted trajectory. In every trajectory, we calculate the number of nodes that satisfy Equation (5.4) to determine if a node can be successfully visited. This gives us an updated set $\vec{\mathbf{T}}'$ which only consists of nodes in the trajectories which are feasible. From this updated set $\vec{\mathbf{T}}'$ we then calculate the movement cost-matrix $\mathcal{G}$,

$$\mathcal{G}(\vec{T}_i') = \sum_{j=0}^{q-1} (n_j - n_{j+1})^2. \tag{5.7}$$

After this process, the trajectory from $\vec{\mathbf{T}}'$ with the maximum number of successfully vised nodes is passed to the intervention controller. In the case multiple trajectories successfully visit the same number of nodes, the trajectory which also minimizes the movement cost will be passed to the intervention controller.

The planning strategies generate a set of m possible trajectories presented by $\vec{\mathbf{T}} = [\vec{T}_0, \ldots, \vec{T}_{m-1}]$, where each $\vec{T}_t$ represents an organized list of weed locations in the trajectory, consisting of q_t elements in form of $\vec{T}_t = [w_0, \ldots, w_{q_t-1}]$. Using the following criteria we calculate the success criterion $\vec{\mathcal{C}} = [\mathcal{C}_0, \ldots, \mathcal{C}_{m-1}]$ for each trajectory in $\vec{\mathbf{T}}$.

$$\vec{\mathcal{C}}(\vec{\mathbf{T}}, \rho) = \begin{cases} \frac{1}{q_t} \sum_{r=0}^{q_t-1} \Gamma(n_r, n_{r+1}), & for\ all\ \Gamma \geq \rho \\ 0 & otherwise \end{cases} \tag{5.8}$$

where $\rho = 0.6$ is a cutoff threshold applied to each feasibility score ensuring the planner only considers trajectories with all reachable targets. To incorporate the harmfulness factor, we use the $\vec{\mathbf{T}}$ where $\mathcal{C}_t > 0$, and pick the trajectory with maximum total harmfulness score $\mathcal{K}$ using,

$$\underset{t}{\text{argmax}}\, \mathcal{K}(\vec{\mathbf{T}}); \ \text{ where } \ \mathcal{K}(\vec{T}_t) = \sum_{r=0}^{q_t-1} \kappa\left(\vec{T}_t(r)\right) \tag{5.9}$$

and κ_r is the harmfulness factor, Equation (5.10), of the r-th node in $\vec{T}_t$. This leads to the best trajectory being the one that has the largest number of reachable targets and high-priority weeds.

5.6 Bio-Diversity-Aware Plant-level Treatment

Weed harmfulness is influenced by aggressiveness, competition for water, and leaf size. Species like barnyardgrass, pigweed, and lambsquarters can significantly outcompete crops for light, water, and nutrients, reducing yield and quality [125]. In moisture-limited environments, this competition becomes more critical. Understanding these factors is essential for effective weed management, recognizing that not all weeds are detrimental; some can enhance soil fertility and moisture retention. An example scenario is depicted in Figure 5.7 where dicots are considered beneficial.

Therefore, robots must recognize and adapt to bio-diversity when managing these varied weed threats. In this case, a generalized approach to weed management will not suffice due to the diverse competitive behaviors of different weeds. We propose a novel weeding method enabling BonnBot-I to consider bio-diversity factors by differentiating between weed species and their phenotypic characteristics, assessing their threat levels based on competitiveness factors, and acting accordingly. This ensures not only the well-being and productivity of crops but also resource optimization concerning weeding actions (i.e. spraying).

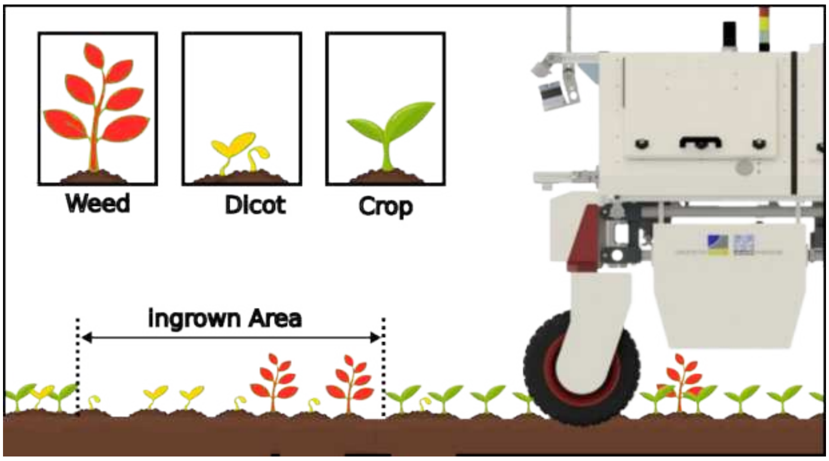


Figure 5.7: Biodiversity in Focus: A field area with low crop (green) density, susceptible to invasive weeds, and the potential role of Dicot (yellow) plants in controlling weed (red) growth.

Relying upon the phenotypic information, we aim to facilitate our robots' understanding of different weeding scenarios in the real field. Using the monitoring technique introduced in Section 3.4 we determine the type, growth stage, and distance of weeds in a cropping area. We introduce the 'harmfulness' factor κ, illustrated in Equation (5.10), for each weed type, ranking them based on their threat to the nearby crops. It is important to acknowledge that currently, there is no definitive method for assessing plant-wise harmfulness effects in actual fields on different crops. However, in this work, we aim to build upon existing heuristics to facilitate incorporating this information into future weeding strategies. Hence, we derived this factor from phenotypic data, which helps our robots discern various weeding scenarios. It ranks weeds based on their threat to crops, using a methodology inspired by Dentika *et al.* [38], who proposed a similar concept for pathogen-host and disease risk imposed by weeds on the crops, and was derived from comprehensive on-site experiments and is formulated below:

$$\kappa(w, p) = \frac{\alpha_w \times \beta_w}{\alpha_p \times \Delta(w, p)} \tag{5.10}$$

In Equation (5.10), the harmfulness effect $\kappa(w, p)$ of weed w on plant p is calculated where, α_w and α_p denote the size of weed w and crop p in mm^2, respectively. This is reflecting the competitive relationship between a weed and its adjacent crop, based on their sizes. Also, β_w is the harmfulness factor of each specific weed category, noting that certain weeds should be eliminated from the field regardless of their size or specific location. Furthermore, the $\Delta(w, p)$ is the Euclidean distance between crop p to weed w. This acknowledges that weeds situated far from crops may not pose a significant threat, thus requiring no intervention. This approach promotes eco-friendly interventions by prioritizing the most harmful weeds, thereby enhancing biodiversity and resource conservation.

5.7 Experiments

In the following, we demonstrate the results of several experiments accomplished in simulation and real environments to show the versatility and robustness of BonnBot-I with its novel weeding implement and our monitoring and intervention approach. We consider that the robot moves with constant speed $\gamma = 0.5m/s$ along a crop row with weed density of λ weeds/m^2 and we set the velocity of the linear actuators to $\vartheta = 5m/s$. All experiments are conducted using 4 linear axes.

5.7.1 Weeding Planning Real-Time Performance

To evaluate the performance of the planning approach for the weeding tool explained in Section 5.5, we use our native Python simulator (Section 3.2) specifically designed for this purpose. We investigate our approach's performance using two types of crop-row models: simulated rows of crops and weeds and real field models (introduced in Section 3.5.6) created based on evaluation rows datasets.

We initially evaluated the performance of the Brute-Force planner against the nOTSP planner. We found that the results were similar in trivial cases with a limited number of targets ($\mathcal{N} \leq 10$). The computational expense of the Brute-Force method increases almost exponentially when there are more than four targets as it has a run-time complexity of $O(n!)$ compared with $O(n^2 2^n)$ of graph-based nOTSP. This is a significant problem for weeding applications where real-time performance is important. In the case where we see only ten weeds in the planning region, the Brute-Force approach requires 3.7s compared to nOTSP which requires only 266μs. This is a prohibitive quality of the Brute-

Force approach and for our two weeding experiments we employ the nOTSP technique.

5.7.2 Planning on Simulated Crop-Row Models

In this experiment, we evaluate the performance of different weed densities and a different number of linear axes. This evaluation uses the simulation environment introduced in Sec. 3.2, allowing us to control the experimental parameters. The field parameters are set to 3 crops-rows in a single lane with a length of 20m. To fully analyze the performance of our approach we vary the weed density such that $\lambda = [3, 5, 10, 20, 40]$ represents the weeds per m^2. Finally, to outline the benefit of having multiple linear axes we show the performance for $\mathcal{H} = [1, 2, 4, 8]$.

The results for this simulated experiment are summarized in Fig. 5.8, where we provide the comparison of weed density to the percentage of missed targets. From this figure, it is evident that increasing the number of linear axes has an obvious impact on results, with 8 axes performing better than all others. We see from this that even with a distribution of 40 weeds (the hardest case) the worst performing 8 axes system, Distance-Based, achieves a loss of $\sim 15\%$. Overall, the Distance-Based approach routinely performs worse as the weed density increases, this is particularly evident as the number of heads increases. This will be further evaluated in the next evaluation, but as the distribution grows the intervention heads need to travel further to meet the demands of the planner, this movement can negatively impact the capacity of the intervention head to reach the next

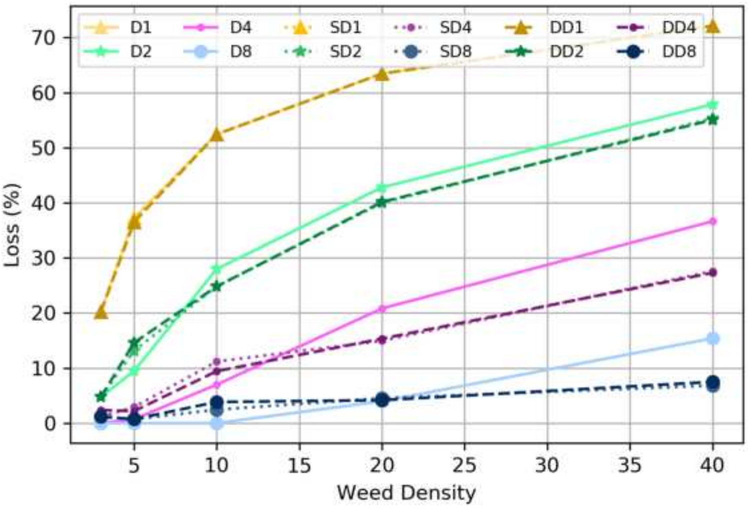


Figure 5.8: The results of the nOTSP weeding planner. The loss represents the number of missed weeds. We evaluate 5 different weed density models (x-axis); and three different planning types: Dist.-Based (D), Dyn.-Div. (DD), and Sub-Div. (SD); and intervention heads $\mathcal{H} = 1, 2, 4, 8$, yellow, green, purple and blue, respectively.

weed. Finally, the two division-based methods appear to perform at a commensurate level across all weed densities in our simulations. We attribute this to the wide distribution of weeds negating the impact of the dynamic approach, meaning, weeds generally appear across the entire lane rather than concentrated in a specific region. Overall, this shows the validity of our planning methods for weed intervention in a uniformly distributed pattern.

5.7.3 Planning on Real Crop-Row Models

To further evaluate the performance of our three planning approaches, we use real weed distributions captured in the fields within the weeding simulator. This information is obtained from the crop monitoring approach outlined in Section 3.4 and aggregated into a simulated row. We perform this on the evaluation rows of the SB20, CN20, and SB21 datasets where we have four different weed distributions: low (CN20), moderate (SB20-S1), high (SB20-S2), high (SB21-S1), and very high (SB21-S2).

The results of our three planning approaches are displayed in Table 5.1, in this experiment we only use four intervention heads as this accurately evaluates BonnBot-I performance. Table 5.1 displays two metrics, first, the percentage of missed weeds, and second, the mean and standard deviation of the distance moved by the axes.

For the percentage of missed weeds, we see that the SB20-S2 with a high density of weeds is the most difficult to intervene on, this is somewhat expected due to the heavy distribution of weeds. However, the simple Sub-division approach outperforms the other two approaches for this distribution which is unlike the shown results in Figure 5.8 for a density of 10 per m^2. The poorer performance of the two other approaches can be attributed to the distribution of the weeds in a real crop row. We performed the Chi-squared test [111] on the sections of the data from the rows and found that the distribution of fields was not uniform, hence increasing the complexity. This performance is mirrored through the other rows,

Table 5.1: The rate of Loss (%) and average traveled distance (m) of interventions heads in real-world weeding scenarios.

	Sub-Div. (SD)		**Dist.-Based (D)**		**Dyn.-Div. (DD)**	
	(%)	(m)	(%)↓	(m)↓	(%)↓	(m)↓
CN20	**0.0**	2.7±0.2	4.3	2.7±2.8	3.4	4.0±3.9
SB20-S1	**0.0**	1.4±0.2	2.3	1.5±0.6	**0.0**	1.0±0.8
SB20-S2	**11.9**	10.1±0.9	19.8	4.7±3.5	13.5	5.0±1.8
SB21-S1	17.2	6.1±5.3	23.5	3.9±3.2	**17.1**	7.1±2.6
SB21-S2	**38.2**	13.4±8.2	48.2	4.1±1.9d	38.9	9.3±3.1

where the static Sub-division approach achieves higher scores. However, once we reduce the weed density (CN20, SB20-S1) we are able to achieve a percentage of missed weeds close to zero for all approaches.

Our final evaluation is based on the movement requirements of the different planning approaches, where a value closer to zero is desired. While static Sub-division achieved the best performance for the percentage of missed weeds we see the negative aspect of this approach here. Both the dynamic division and distance-based approaches can achieve considerably better results in the traveled distance, in the case of SB20-S2 both other approaches move half as much as the static Sub-division approach. For SB21-S1 and SB21-S2, the static Sub-division approach demonstrates lower weed missed percentages, highlighting its effectiveness in high weed density situations. However, in terms of traveled distance, the SD approach results in significantly higher, which could be attributed to the corresponding weeding performance gain. We will utilize the static Sub-division approach in future evaluations as the nOTSP baseline.

5.7.4 Segment-view Observations vs Rolling-view Observations Models

Here we illustrate the efficiency of the weeding system on real-field models, showcasing the nOTSP approach in two distinct observation modes, Segment-view (baseline) and our novel Rolling-view observations. Our evaluations are applied to the test rows within the three datasets (CN20, SB20, and SB21). Table 5.2 summarizes the difference in planning performance and traveled distance of weeding axes between the two observation models (Segment-view (baseline) and Rolling-view) deployed on the real-field models. Below we briefly describe the results in Table 5.2 in terms of weed density, from low to very high.

Low weed density: both methods demonstrate comparable performance in areas with lower weed density achieving zero loss on CN20 and leaving no untreated weeds in the fields. The Rolling-view model, however, required less

Table 5.2: Weeding performance of BonnBot-I with two observation models Segment-view and Rolling-view denoting weeding loss (%) and traveled distance (m) of the linear axis.

	Weed-density		**Segment-view**		**Rolling-view**	
	-	Avg/m^2	(%)↓	(m)↓	(%)↓	(m)↓
CN20	low	3.1	**0.0**	2.7±0.2	**0.0**	2.0±1.6
SB20-S1	moderate	8.2	**0.0**	1.4±0.2	**0.0**	1.9±0.9
SB20-S2	high	15.4	11.9	10.1±0.9	**6.4**	4.1±2.1
SB21-S1	high	22.3	17.2	6.1±5.3	**14.1**	2.8±2.6
SB21-S2	very high	81.2	38.2	13.4±8.2	**36.5**	5.2±3.1

travel distance compared to the Segment-view. However, the variance in the traveled distance of the weeding axes seems to be lower when using Segment-view planning. This suggests a more balanced distribution of targets across the different implements. **Moderate weed density:** Similar to low density, both models achieved 0.0% loss, with no noticeable difference in weeding efficiency. However, the Rolling-view model had a slightly higher travel distance ($1.9 \pm 0.9m$) than the Segment-view ($1.4 \pm 0.2m$). **High and very high weed density:** In this scenario, the Rolling-view model outperformed the Segment-view model. The Rolling-view achieved only a 6.4% loss compared to the Segment-view's 11.9% loss. Additionally, the Rolling-view required a substantially lower travel distance than the Segment-view, showing the efficiency of the Rolling-view method in more complex situations. Overall, the Rolling-view planning method surpasses the baseline with an average absolute improvement of 3.5% over all the field models with a maximum and minimum absolute improvement of 5.5% on SB20-S1 and 1.7% on SB21-S2, respectively.

5.7.5 Bio-diversity aware weeding operation

The bio-diversity-aware system should accurately differentiate between crops and weeds. This includes detecting beneficial dicots and only intervening where it is essential for crop health while preserving and promoting biodiversity. This weed classification approach is particularly crucial in challenging scenarios. These scenarios can include high weed density, restricted robot velocity, restricted linear axis speed, or herbicide usage limitations. Hence, the system must manage certain losses while achieving optimal performance. To test our system's resilience and capability under extreme conditions, we devised an experiment where the robot operates at a speed two times the normal $\gamma = 1.0m/s$.

In this analysis, we utilized the field models from our previous experiment but with two modifications. First, we designate two priority levels where a low-priority for weeding is given to potentially beneficial dicots and a high-priority is given to all other weeds. To represent the priority levels we give a weight of 0.1 and 1.0 for low- and high-priority respectively. Second, the occurrence of high-priority weeds was set to be one-tenth of that of low-priority. Consequently, this setup required the bio-diversity-aware system to adjust its weeding trajectories to prioritize the less frequent, high-priority weeds, even if it meant potentially overlooking some of the low-priority weeds.

In the following, we compare the bio-diversity-aware (Bio-Div.) approach with a baseline (not-bio-diversity-aware) method under extreme conditions on the same real-field models used in the previous experiment.

The weeding loss for the Bio-Div. the approach is very similar to the baseline. It leaves no untreated weed in the fields, indicating its comparable effectiveness

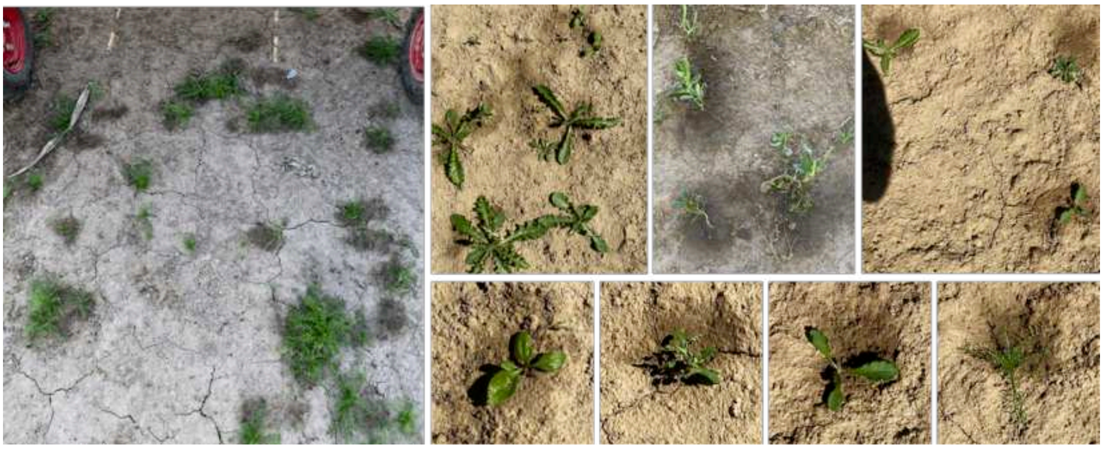


Figure 5.9: Real-field intervention examples; Different spray footprints of BonnBot-I weeding operation in campus CKA of the University of Bonn, with different weed densities and various weed types.

in scenarios with low to moderate weed densities. However, in cases of very high weed densities (SB21-S1, SB21-S2), there is a minor ($< 2\%$) increase in weeding loss with the Bio-Div. approach. This suggests a small trade-off in performance at extremely high weed densities. The average traveled distance of axes using both observation methods is generally similar or marginally higher ($< 0.2m$ on average) for the Bio-Div. approach across all field models, reflecting the method's thoroughness in targeting specific weed priorities and achieving bio-diversity considerations. Additionally, to enhance our evaluation of the operation's specifics, we detailed the treatment percentage for each weed type individually and their results are presented in Table 5.3.

When considering the low-priority weeds, the Bio-Div. method achieves slightly reduced performance when compared to the baseline. On the other hand, this reduced performance could be interpreted as a positive point, where keeping low-priority weeds in the field could benefit the ecosystem and align with bio-diversity purposes. However, the benefit of this approach is evident in the ability

Table 5.3: Weeding performances (treatment percentage) of BonnBot-I with baseline weeding and Bio-diversity-aware schemes on real-field models.

Field Model	**Weed 1** (Low Priority) (not-Bio-Div.) (%)↑	**Weed 1** (Low Priority) (Bio-Div.) (%)↑	**Weed 2** (High Priority) (not-Bio-Div.) (%)↑	**Weed 2** (High Priority) (Bio-Div.) (%)↑
CN20	73.3	73.3	56.6	56.6
SB20-S1	**47.5**	44.3	56.0	**60.2**
SB20-S2	**42.5**	40.5	53.5	**63.5**
SB21-S1	**18.0**	15.2	19.1	**38.8**
SB21-S2	**7.0**	5.1	7.0	**17.4**

of the Bio-Div. approach to achieve stronger performance when targeting higher-threat weeds. Our approach is able to improve performance on all but the lowest weed distribution (CN20) where performance is identical. This illustrates the Bio-Divapproach's effectiveness in prioritizing and managing high-priority weeds.

5.7.6 Real-World Intervention Performance

In [5] we only evaluated the weeding performance on recorded data and considered non-overlapping segments of observations for running the experiments. To further evaluate the whole system's performance, we deployed BonnBot-I in a series of unseen fields with different weed distributions, illumination conditions, and cultivars. The experiments were conducted on various days, under a mix of weather conditions including partial cloudiness and sunshine, and on different types of soil ranging from solid and compact to relatively muddy. Additionally, the plants tested were at various growth stages, from the two-leaf stage up to the eight-leaf stage. The field experiments with the real robot were conducted at the CKA, covering nearly 100 square meters of sugar-beet field. This evaluation outlines the most important metrics associated with the operation of the robot. These metrics include the number of missed weeds, the number of partially treated weeds, and the number of accurately treated weeds. We quantitatively evaluated our performance on two different weeding scenarios: crop-weed; and weed-only.

During all evaluations, we used the Rolling-view observation model enabling more accurate planning in real-world scenarios. An example of this planning approach is depicted in Figure 5.10 where only three linear axes are utilized (indicated by the three colors). The weed-only field model was used to evaluate our intervention approach independently of vision system failures (i.e. classifying a weed as crop or vice versa). Our quantitative analysis is based on visually interpreting the footage captured on a camera mounted at the rear of BonnBot-I. Figure 5.9 demonstrates the footprint of weeding interventions in the real fields.

The trial fields contained regions with low (< 5) to high weed (> 10) densities, therefore offering challenging scenarios for testing BonnBot-I weeding capabilities. In total our two fields contain 1038 weeds, the sugar-beets in the crop-weed fields had plants in the four- to six-leaf stage (approximately three to six weeks old). Table 5.4 summarizes the performance of BonnBot-I in both weed-only and crop-weed regions. Initial evaluations in the crop-weed region showed, that BonnBot-I has treated 23 crops falsely which counts as a total of 4% of visited crops in the field. Furthermore, of the total weeds in the fields, we accurately detected 886. Of these detected weeds BonnBot-I did not perform an intervention on 121 of them.

Consequently, the weeding loss attributable to planning or intervention con-

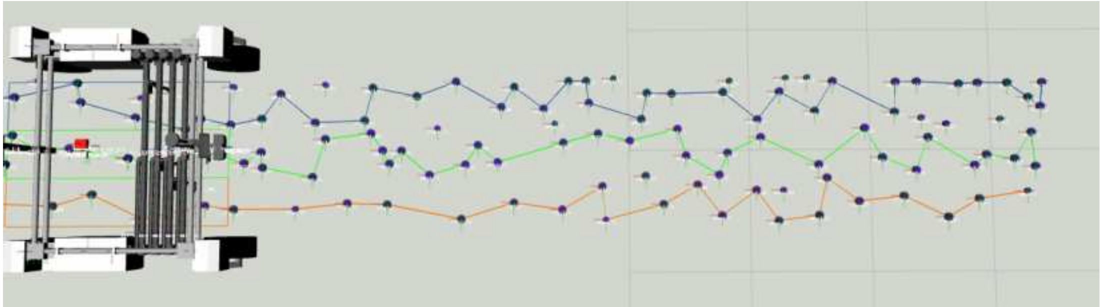


Figure 5.10: BonnBot-I Simulation; An example field with weed density $\lambda \approxeq 12$ with planned trajectories of three linear axes in different colors (blue, green, and orange). All dots that lie along one of the lines are weeds that will be treated, while all the dots that do not lie upon a line cannot be treated.

Table 5.4: Real-world weeding performance of BonnBot-I in weed-only and crop-weed regions.

	Crop	**Weed**			
Field Model	False Hits	Total Weeds	Acc. Hits	Par. Hits	Missed
Weed-Only	-	473	183	192	98
Crop-Weed	4%	565	177	213	175

straints stands at 11.6%. This loss could be addressed by fine-tuning planning strategies, adding more weeding axes, increasing the velocity of the linear axis, or decreasing the robot's linear velocity. The remaining 152 weeds $(1038 - 886)$ were missed because of systemic issues, notably the vision system's performance under difficult real-time and challenging lighting conditions. Hence, the overall loss of the system added up to 26.3% of the total number of weed plants in the fields (including the detected-and-missed and not-detected instances).

Further examination showed that from the total of 98 missed weeds in the weed-only region, a total of 39 plants remained untreated solely because of the vision system's inability to detect them. Similarly, in the crop-weed region, 113 weeds were missed due to vision system failure highlighting the limitations of the vision system and the challenges encountered in real-field conditions. This loss could be reduced by improving the vision system and using more advanced DNN architecture in the future.

Weeds were successfully treated in the real-world field trials of BonnBot-I 765. Of the treated weeds, on average 47% were considered to be treated in a highly accurate manner, 183 and 177 weeds in weed-only and crop-weed regions, respectively. This means the spray footprint was centered on the weed. The remaining 53% were treated successfully but with less accuracy, with an offset of nearly $2cm$. They are considered to be partial treatments as they still cover a large proportion of the weed. This reduced performance is likely linked to the variable conditions encountered in real-field settings. Factors such as slight weather changes, like unexpected wind gusts, can alter spray distribution by a few centimeters or move plant leaves, thereby impacting the accuracy of weed detection and treatment. While several factors caused these errors, we still consider this a successful treatment of the weeds in the field.

5.8 Conclusion

In this chapter, we introduced several novel precision weeding approaches based on the BonnBot-I platform. These approaches were evaluated in both real-field conditions and simulated environments. The simulated environment accurately mimic real-world weed distributions, leveraging data from our field monitoring techniques running on real-fields at CKA campus of the University of Bonn.

Experiments conducted in these simulated fields demonstrated the effectiveness of our proposed workspace division techniques, showing a significant reduction in movement ($10m$ compared to $5m$) when compared to distance-based target assignment methods. We also presented an advanced planning method using a rolling-view technique, which led to an average absolute performance improvement of 3.4% compared to segment-view planning. Additionally, for the first time, we explored the concept of biodiversity-aware weed management and assessed its practicality in real-world scenarios. Our approach not only improves the precision and effectiveness of agricultural robots but also highlights the importance of incorporating ecological considerations into crop management.

Furthermore, our real-field experiments with actual robots demonstrated that the BonnBot-I weeding strategy could be effective with only an 11.66% loss due to planning or intervention constraints. In conclusion, our research indicates that performance variations in weeding technologies are often influenced by dynamic real-field conditions, such as unexpected weather changes impacting weed detection and treatment accuracy.

This work and the development of the BonnBot-I enabled us to further exploration of plant-level interventions on arable farms. Additionally, it has motivated us to create a more intelligent approach for controlling weeding mechanisms and interacting with the environment. Future efforts will focus on enhancing these systems' adaptability to environmental factors by developing more robust and responsive vision systems, ensuring consistent and effective weed management across diverse agricultural settings. As a result, we have devised a method to learn intervention concepts using deep reinforcement learning (RL). In the next chapter, we will delve into the architecture and our approach for modeling precision plant-level weeding as an RL problem, which can integrate various modalities of information and effectively plan interventions for different scenarios.

Chapter 6

Learning to Perform In-Field Intervention

In this chapter, we introduce OptimWeeder a novel reinforcement learning (RL)-based approach to control precision weeding operations on BonnBot-I introcuded in Chapter 3. The proposed method is adapted from the well-known proximal policy optimization (PPO) method [95]. This method learns to predict the trajectory and move the intervention heads on the weeding tool of BonnBot-I in an efficient manner.

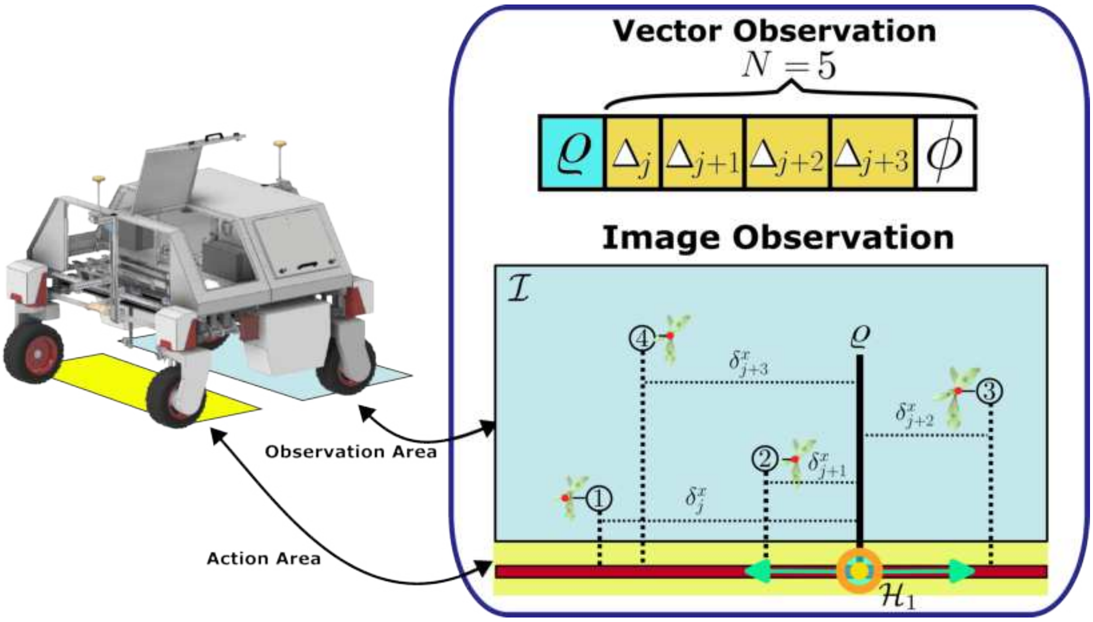


Figure 6.1: OptimWeeder Environment; observations from the BonnBot-I platform including four weed (circled ids), weed position, weeding-tool, and robot kinematics, agent current position ρ are translated to an RL-agent $\mathcal{H}_1$ observation space and action space to learn a policy to control weeding nozzles. The OptimWeeder Environment allows the agent to observe the environment as an image or through vector representations which enables training an effective RL-based weed control policy.

In Chapter 5, we investigated classical planning methods. We deployed these on BonnBot-I to control the motion of the weeding heads, which were able to intervene on plants precisely and accurately. However, these classical methods, such as brute force and nOTSP lack flexibility and scalability in diverse weeding scenarios meaning they often fail in real-world agri-robotic situations. This is most evident when there is a need to consider modalities such as plant-specific actions (bio-diversity aware weeding).

To achieve a more flexible approach we model the precision weeding problem as an RL problem by considering th real kinematics of BonnBot-I. RL can efficiently adapt and scale through the reward function design, bypassing the need for complex heuristics, as seen in classical approaches. Another benefit of our RL approach is that it excels in managing partial observations and adapts quickly in real-time. Coupling our RL approach with BonnBot-I's independently movable and controllable weeding heads (depicted in Figure 6.1), allows precise application of herbicides. This ensures BonnBot-I is able to directly target weeds while minimizing herbicide usage and avoiding crop damage.

Another key benefit of our RL based approach is the ability to smootly transition between single- and multi-agent schemes by using a decentralized policy execution protocol. We conducted comprehensive experiments in a simulated environment using real-world field models of sugar beet and corn fields, encompassing a wide range of weed densities and distributions. The results demonstrate that our novel RL-based weeding approach outperforms state-of-the-art nOTSP planners, leaving fewer untreated weeds in the fields. Additionally, our novel biodiversity-aware RL-based method outperforms an nOTSP-based biodiversity-aware planning approach.

As introduced in Chapter 3, the spray nozzles are independently movable and controllable modules, allowing precise application of herbicides directly. We model each linear axis as an independent RL-agent, allowing active interventions in real time as the robot navigates the field. Our results demonstrate that the proposed method effectively learns control policies for managing biodiversity-aware weed removal in realistic field conditions. OptimWeeder integrates multi-modal weed observations into the planning process, enabling more informed and biodiversity-conscious weeding decisions. To achieve this, we have made significant developments to enhance the capabilities of BonnBot-I. The key contributions of this work are as follows:

1. The development of the first OpenAI-Gym compatible simulation platform specifically designed for precision weeding applications.

2. A novel reinforcement learning-based strategy for precision weed management based on real kinematics of our self-built platform BonnBot-I.

3. The introduction of the first biodiversity-aware precision weeding scheme using reinforcement learning.
4. Validation of the proposed approach through synthetic and real-field models in simulation.
5. The release of the weeding gym environment as a standardized benchmark for evaluating precision weeding approaches.

This final contribution chapter focuses on our approach to RL for optimizing plant level interventions. Following the introduction and literature review, we introduce the RL framework developed for BonnBot-I in Section 6.2, which enables the robot to learn and adapt its intervention strategies based on plant priorities.

Subsequent sections delve into observation modeling (Section 6.3), action space design (Section 6.4), and the reward function (Section 6.5) in detail. Finally, in Section 6.6, we present experimental results and compare them directly to the state-of-the-art methods discussed in Section 6.7.

6.1 Related Works

Agricultural robotics has undergone a revolution, driven by cultural expectations and resource allocation to achieve sustainable solutions [122]. Reducing labor costs is crucial for farmers, as it represents a significant business expenditure [2]. Improved automated monitoring techniques now enable more robust decision management schemes, including interventions that significantly reduce environmental impacts, such as soil compaction. Additionally, a wide range of weeding implements have been used to deal with different types of weeds in various farm conditions, including organic and non-organic fields. Examples include mechanical hoeing systems [19, 27] used in organic fields, precise sport-spray systems [119], and electrocution [10] and laser-based [121] end-effectors. These novel approaches offer different benefits and restrictions. They enhance farming operations by reducing agrochemical use and directly improving crop yield [104]. However, most of these methods rely on handcrafted and classical techniques, which lack the flexibility to incorporate multi-modal information (weed type, harmfulness factor, distance to crop, etc) or biodiversity considerations in their intervention strategies. Here, we build upon our advanced biodiversity-aware scheme in [6] by integrating an RL-based learning process for obtaining the optimal intervention scheme in various weeding scenarios.

Early RL methods leveraging the power of deep learning as function approximators were initially applied to learning policies solving challenging games in controlled simulations [75, 96, 114]. However, a long-standing challenge in RL research is transferring policies trained in simulation to real-world scenarios without drastic performance loss [37]. This issue is particularly prominent in real-world robotic control tasks, as the policy learned in the simulation must be robust to sensor and actuator noise, system delays, and changing environmental conditions.

More recently, with the advent of advanced and versatile RL algorithms [95, 50] combined with hardware-accelerated realistic physics-based simulators [37, 93], various challenging robotic control tasks have been solved with RL approaches [57, 62]. These approaches have even been able to outperform well-known non-RL control methods [103]. Kaufmann et al. [57] showcase professional human-level control performance in the highly agile drone racing task, combining training a policy from large amounts of simulated racing data with a sophisticated simulator design closely approximating real-world actuation and perception noise. Song et al. [103] empirically show that such RL approaches to drone racing might outperform classical model-predictive control methods. This is due to RL facilitating the algorithm design by shaping outcome-based reward functions that directly reflect the mission goal. Similarly, Hwangbo et al. [56] propose an RL-based system to train locomotion policies for legged robots in simulation that are directly deployable in real-world scenarios without retraining. Lee et al. [62] extend this approach to enable real-world deployment in challenging, e.g., muddy or snowy terrains which were not encountered during training in simulation. Other domains RL-based robotic control was successfully applied to include point-goal navigation with household robots [116] or dexterous hand control for manipulation tasks [124].

In our previous chapter (Chapter 5), we presented a method to control BonnBot-I's multi-head weeding tool using a variant of the traveling salesman problem, called the nOTSP approach. A significant challenge within this system involves incorporating additional modalities (weed type, harmfulness factor, size, etc) into the decision-making process. To address this, we introduce a more flexible weed control strategy. Exploiting the benefits of RL methods, we propose an RL-based algorithm trained on a custom-developed OpenAI Gym simulator (see Section 3.2) that accurately replicates the kinematics of BonnBot-I. In contrast to previous weed control methods, our framework incorporates selective interventions targeting specific weed species based on priority by integrating this information into the observation. These features enable the algorithm to adapt its behavior to biodiversity considerations, prioritizing interventions based on the most harmful weeds and minimizing unnecessary interactions.

6.2 Problem Definition with RL

A key aspect of our approach is the encoding of real robot constraints to ease the deployment on real conditions. The observation and the reward functions for

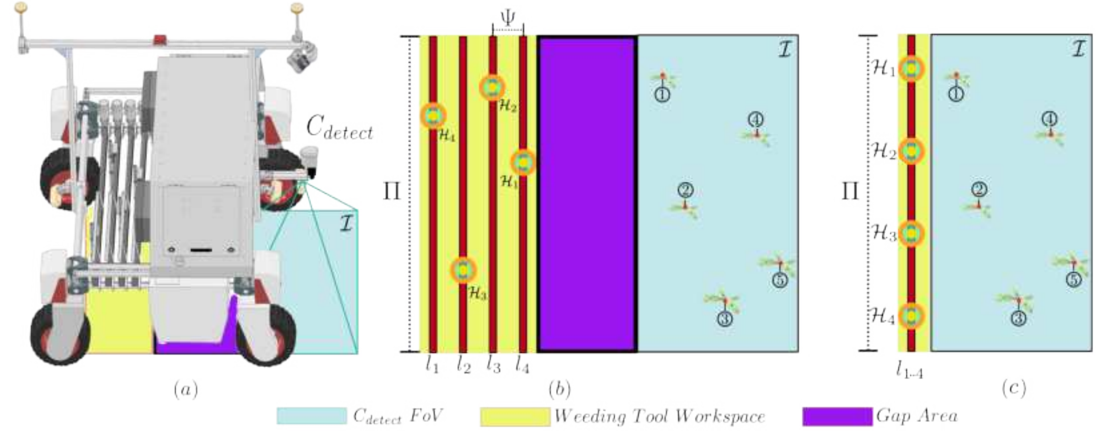


Figure 6.2: BonnBot-I Weeding Schematics; (a) A top view of BonnBot-I with observation space of C_{detect} in light-Cyan and the workspace of weeding tool in yellow is depicted. Weeding Tool and Observable Areas: (b) Actual kinematics model of weeding tool w.r.t the observations area (Cyan windows) with four linear axes ($l_{1\ldots4}$) of equally spaced Ψ. (c) Gym environment model of weeding tool lining up all nozzle ($\mathcal{H}_{1\ldots4}$) on a single bar at the bottom of the observable area.

learning an optimal weeding strategy, and a simulation environment specifically designed for precision weeding application.

We define a weeding scenario which consists of a set of weeds (targets) denoted by w and tools (nozzle heads) $\mathcal{H}$ that visit/spray the weeds. A nozzle is given a set of q_t weeds $\mathcal{W}_t = \{w_0, w_1, \ldots, w_{q_t-1}\}$ where w_j is a two-dimensional vector consisting of the x-y stem position in the robot's frame at time t. Let $\Delta_t(\mathcal{H}_i, w_j)$ denote the Euclidean distance between the i-th tool and the j-the weed. A valid trajectory visits n targets and every weed must be visited only once while also minimizing the distance traveled. Therefore, the nOTSP target function, Equation (6.1), considers a trajectory of length n as $\vec{\mathbf{T}}_t = \{w_j, w_{j+1}, \ldots, w_{n-1}\}$, where each $w_k \in \mathcal{W}_t$ and n is constrained to be $n <= q_t$

$$\min_{n} \sum_{j=0}^{n-1} \Delta(w_j, w_{j+1}) + \Delta_{ij}(w_j, w_0). \tag{6.1}$$

To obtain valid observations we used data from three real-world fields using BonnBot-I. These fields, based at Campus Klein-Altendorf (CKA) of the University of Bonn, were captured over two years and represent a range of weed densities, ranging from low (CN20), moderate (SB20-S2), high (SB20-S1, SB21-S1), and very high densities (SB21-S2). More information about these publicly available datasets can be found in Section 3.3 [4, 5, 6]. As introduced in Chapter 3, BonnBot-I has modular and repeatable weeding tools with interchangeable end-effectors, allowing tool changes based on soil and weed conditions. The design involves independently controlled linear actuators placed at 0.72m above the ground, providing a working space of 1.3m × 0.36m. There are four linear actuators, each with a length of $1.3m$. Currently, BonnBot-I has high-speed spray valve heads installed, with an on-off time of $\sim 12ms$. Individual adjustable valves are utilized to control the droplet size from the spray nozzles which can be set to have a spray footprint between $0.02m$ and $0.13m$. Similar to Chapter 5, the robot linear speed is set to $\gamma = 0.5m/s$ with weed density λ (weeds/m^2) provided from real field models, and we set the velocity of the linear actuators to $\vartheta = 5m/s$. The field models, extracted from the crop monitoring procedure, outlined in Section 3.5.6, are structured into a representative crop-row format. To achieve a correct hit (of the weeds) we set the spray footprint to $4cm$ meaning that a resulting hit of greater than $2cm$ from the center is required.

The choice between single-agent and multi-agent systems is pivotal. In a multi-agent framework, each linear axis assumes the role of a distinct agent, offering flexibility, especially when varying the number of linear axes and their workspace. However, challenges arise from this approach, notably due to non-stationary environments where changes depend on other agents actions. Furthermore, revealing the contribution of individual agents to the global reward and determining the credited assignment problem leads to greatly increased complexity. Hence, we decided to use a single-agent model and replicate them for each tool to match the kinematics of the BonnBot-I weeding tools.

6.3 State and Observation Space

In a weeding scenario, the robot moves at a constant speed through the crop rows, detecting and tracking each plant in real time. An accurate representation of the weeds/targets is required to plan appropriate trajectories for each wedding axis, enabling precise plant-level interventions. To create realistic observations and effectively train the RL agent, we rely on real-field models as introduced in Section 3.5.6. This observation model must capture the robot's critical kinematic parameters and the targets' key characteristics to ensure precise weeding actions. We have a partial observation of the environment, precisely reproducing views of the robot camera (C_{detect}) represented in vector and image-based forms. The gap between the detection area (C_{detect}) and the workspace of the weeding tool, as illustrated in Figure 6.2-a, means that by the time an action is performed for segment T_0, segment T_2 has already been observed due to the robot's constant velocity. This can be advantageous in situations where weeds are unevenly distributed or when high-priority weeds appear in later sections. However, in the gym environment model, we control the gap between the detection and working areas to be minimal. This could potentially limit the number of visible targets at any given time, making the observation vector simpler and reducing complexity, which aids in learning a more efficient weeding strategy. Hence, we sets the nozzles in a position directly after the observation area.

We consider the Markov decision process (MDP) as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P})$, where $\mathcal{S}$ represents the set of possible states, $\mathcal{A}$ denotes the action space, $\mathcal{R}$ is the function that assigns a reward to each state, and $\mathcal{P}$ characterizes the probabilities of transitioning between states. The stochastic policy of an agent is symbolized by $\pi(a|s)$, indicating the probability of sampling action $a \in \mathcal{A}$ when in state $s \in \mathcal{S}$. The RL objective is to identify the optimal policy that maximizes cumulative rewards over time through sequential action selection based on the evolving environment state, $R_t = \mathcal{R}(a_t, s_t, s_{t+1})$, where t is the time step of the epoch and s_t is the initial state drawn from a predefined distribution $\mathcal{P}_0$. In each state (s_t) the agent receives a representation of weeds visible in the observation space and can take only a single action (a_t) with the type of, either nozzle movement or weed intervention.

The developed training environment provides two types of agent observations: vector-based and image-based. Image-based observations which mimic the perspective of a detection camera positioned in front of the BonnBot-I, as shown in Figure 6.3, are only used for visualization purposes. The vector-based observation $\vec{\mathcal{S}}$ is set to a fixed length comprising two parts, as shown in Figure 6.4:

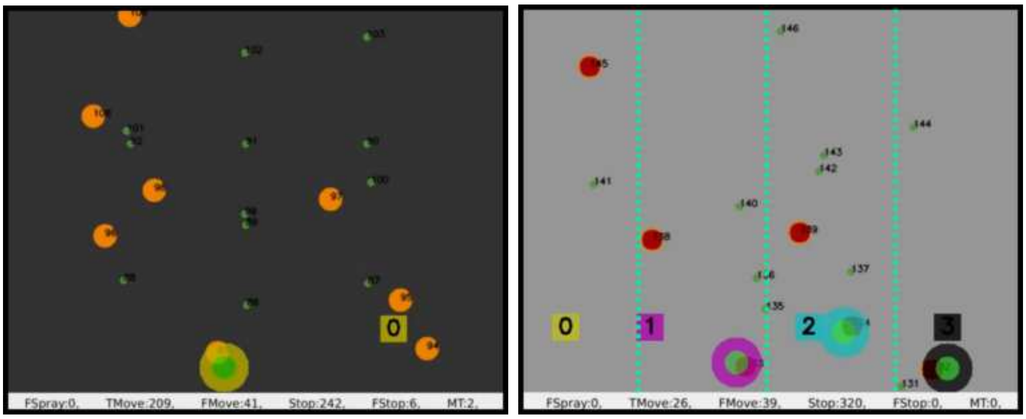
- The agent's state in cyan $\vec{\mathcal{S}^a}$, the current intervention head position ϱ in its linear workspace Π.
- The weeds' states in orange and green $\vec{\mathcal{S}^t}$, a list of float values that represent the relative distance of the i-th intervention head to the visible weeds underneath the robot and their priority information ∇.

In the following, we elaborate on how we develop a new representation of the agent observation for the weed control problem.

Agent State: The agent is defined based on the model of linear axes in the weeding tool, hence it only moves laterally (y-axis) while the whole robot carries it in the x-direction as shown in Figure 6.2-b. The first element in the observation vector denotes the nozzle head's position ϱ in its linear workspace Π, shown in Figure 6.2-b. We use zero-centric positioning schemes (zero standing in the middle of the linear axis) to map nozzle head movements in $\vec{\mathcal{S}}$. This approach centers the observation reference at zero, which lies at the midpoint of the workspace Π (refer to Figure 6.2-b). This division creates an evenly split workspace into positive and negative regions (ϱ ranging from -1.0 to 1.0).

Target's State: We consider the relative position of target weed w_j w.r.t the nozzle head $\mathcal{H}_i$ in the observation environment, which is parameterized based on Euclidean distance Δ and cast in a proper domain using *arctan2* kernel, shown in Equation (6.2).

$$\Delta'_{ij} = \arctan 2(\sqrt{(\mathcal{H}_i^x - w_j^x)^2 + (\mathcal{H}_i^y - w_j^y)^2},\ \lambda) \tag{6.2}$$


0
FSpray:0, TMove:209, FMove:41, Stop:242, FStop:6, MT:2,
0
1
2
3
FSpray:0, TMove:26, FMove:39, Stop:320, FStop:0, MT:0,

(a) Single-Nozzle

(b) Multi-Nozzle

Figure 6.3: BonnBot-I weeding Gym environment; (a) Single-nozzle (yellow) setup, (b) The setup with Multi-agents (four nozzles: yellow, pink, cyan, and black squares) in a field with crops (green circles) and weeds (red circles). The spray footprint circles with a central green core indicate successful intervention.

where $\lambda = 0.02m$ is a constant value to evaluate the accuracy of interventions. To make a fixed-dimensional observation vector, we define the maximum number of observable targets to appear in the observation vector. The targets that get hit with a spray disappear from the next observation vector while remaining in the visible region. It can help emphasize certain features or reduce noise, making the observation more informative for the model.

Normalization: We normalize the target states, Δ_{ij}, to range between -1 and 1 using Equation (6.3). This process better encodes relative differences of the target states and empirically we found this leads to faster and more stable learning by ensuring that no feature dominates due to its scale.

$$\Delta_{ij} = 2\frac{\Delta'_{ij} - \min(\Pi)}{\max(\Pi) - \min(\Pi)} - 1. \tag{6.3}$$

Sort: This attribute specifies the reference entity in the observation space. It determines the sorting procedure for observations in a vector. It could be chosen between (I) randomized, (II) closest target in the X-axis, (III) closest target in the Y-axis or first-in-first-out (FIFO), and (IV) closest target based on 2D Euclidean distance. We use the FIFO sorting method, as we found it empirically more compatible with real weeding scenario.

Biodiversity Priority: The impact of weeds on the crop varies depending on the species and environmental conditions. Integrating biodiversity-aware factors into the observation vector for the RL agent equips it with the capability to learn more sustainable intervention strategies. To enable the RL agent to learn such policies, a key feature of our approach is to consider the priority ∇ of targeted plants, used to adjust intervention rewards accordingly, as described in Section 6.5. Furthermore, we include priority data and different types of plants

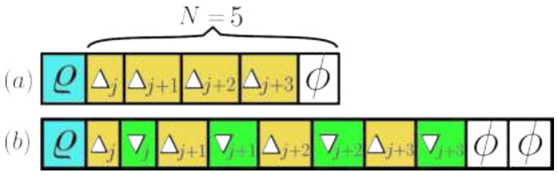


Figure 6.4: Observation vector structure. The case with four targets on the visible segment S_t and BonnBot-Ionly uses one nozzle head. (a) observation vector of the RL-based scheme when the agent only receives lateral distance Δ_{ij} of each visible target, (b) observation vector when the agent perceives lateral Δ_{ij} distance and priority ∇_{ij} of each visible target. The ϕ denotes to constant space older value.

within the observation vector, allowing the RL agent to discern between high-priority and low-priority weeds and thus take actions which tradeoff the number of weeds treated and their priority.

Observation Modality Arrangement: To prepare the observation vector the model stacks characteristics of each visible target in two different observation vectors depicted in Figure 6.4, where (a) shows an equal-priority observation vector with only distance stacked in a vector and (b) shows an example of bio-diversity-aware observation vector grouping 2D distance Δ_j and plant priority ∇_j for each target in the vector.

6.4 Action Space

We introduce three relative motion controllers to allow the agent to perform actions with high precision and flexibility these being: constant, dynamic, and continuous. The proposed action space include three main action types: spray, motion, and stop. Hence, the action space of the defined agents does not control the motor torque or speed; a low-level controller manages it to enhance robustness. Motion of the axes are controlled by their position in a discrete space having a maximum resolution of $1mm$, this matches real linear systems' maximum resolutions too.

Constant: In this model, the agent is limited to making fixed movement steps to the left or right, with a size $\psi = \pm 15$ pixels translating to $1.5cm$ on real linear axis coordinates. The sign of ψ determines the direction of movement: negative for left, positive for right, and zero for remaining stationary. In the constant model, the valid actions are as follows: (spray: 0, stop: 1, move-right: 2, move-left: 3, move-right-and-spray: 4, move-left-and-spray: 5).

Dynamic: Enhancing the agent's flexibility and responsiveness to better deal with complex scenarios where variable step sizes are needed we use a range of predefined motion sizes in the form of a vector $\vec{\Psi}$. In this model, rather than moving with predetermined fixed step sizes, the agent adjusts its motion step size by choosing it from a predefined range of motion $\psi[i] = \vec{\Psi}[i]$.

Continuous: This model permits the agent to execute movements across a continuous range, enabling finer and smoother adjustments. This flexibility is particularly advantageous in complex tasks like unbalanced weed densities across row-crop fields. Hence, the agent can precisely position itself anywhere within the workspace, adapting to changing conditions and requirements for more accurate and effective interventions. In the dynamic and continuous models, the valid actions are as follows: spray: 0, stop: 1, motion: ψ defined in a range of min: -1.0, max: 1.0, min-step: ± 0.01.

6.5 Reward Function

We design a reward function to reflect desirable weed control behavior. The agent gets intermediate rewards for every action in the episode. Besides only a binary feedback system of hit or miss of plants, we found it useful also to consider the area of the weed that has been sprayed or covered as this provides more detailed feedback of precision of the executed action. Hence, after each action, spray footprints are evaluated based on the accuracy of the intervention, considering both true and false sprays. A spray is interpreted as a true intervention only if it hits a weed and false if it lays on empty ground or a crop. The reward function is based on the normal distribution of the distance between the agent and the closest weed in the y-axis is given by,

$$R(s_t, s_{t+1}, a_t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\Delta_{ij})^2}{2\sigma^2}\right) + \zeta\cdot, \mathbf{1}_{cbf} + \Lambda \cdot \mathbf{1}_{mwf} \tag{6.4}$$

which encourages the highest reward for zero distance to the target, and σ controls the spread of the reward distribution. A constant penalty of $\Lambda = -1.0$ is applied to the reward function Equation (6.4) using an indicator function $\mathbf{1}_{mwf}$ for each completely missed target (an untreated weed that leaves the observation and action space). The movements of the agent is determined by their magnitude and direction, either towards or in the opposite direction of the closest target. A strong penalty of $\zeta = -100$ is given for hitting the workspace boundaries (far left and far right of the workspace) integrated into the reward function using $\mathbf{1}_{cbf}$ indicator function. In observations without any targets, we allocate a penalty for any motion or spraying and a reward for not taking any action. To evaluate the stop action of the agent, we consider a reward for stopping within a certain threshold λ close to the nearest target in the y-axis. Stopping falsely, when at least one visible target and nozzle lateral position is not within $\pm\lambda$, also incurs a penalty relative to the distance between the target and nozzle head Δ_{ij} according to Equation (6.4).

Furthermore, we integrate a biodiversity-aware mode within the RL system to enable the prioritization of different crops and weeds. This leads to more informed decision-making and potentially boosting weeding performance in stressed situations, such as when the velocity of the robot is increased. Our reward function incorporates a specific plant's priority/harmfulness factor Equation (6.6) and further explained in Section 5.6. The agent is encouraged to perform an action on high-priority targets, where missing a high-priority weed incurs a significant penalty of $\Lambda = -20$. Similarly, correct movements and stops towards high-priority weeds earn an additional reward of κ relative to the importance of the weeds.

$$R(s_t, s_{t+1}, a_t) = \kappa(w_j, p_k) \cdot \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\Delta_{ij})^2}{2\sigma^2}\right) + \zeta \cdot \mathbf{1}_{cbf} + \Lambda \cdot \mathbf{1}_{mwf}, \quad (6.5)$$

where

$$\kappa(w_j, p_k) = \frac{\alpha_w \ \cdot \beta_w}{\alpha_p \cdot \Delta(w_j, p_k)}. \quad (6.6)$$

The harmfulness factor $\kappa(w_j, p_k)$ of j-th weed w_j on k-th plant p_k is determined by their sizes α_{w_k} and α_{p_k} (in mm^2). The factor β_{w_j} represents the specific harmfulness of each weed category, indicating that some weeds must be removed regardless of size or location. Additionally, $\Delta(w_j, p_k)$ is the Euclidean distance between the crop p_k and weed w_j. In experiments, for simplicity, we assigned weed priorities ∇_j to be equal to $\kappa(w_j, p_k)$, hence the harmfulness of a weed will underscore the importance of prioritized intervention in the weeding process.

6.6 Experimental Setup

To train and evaluate our RL weeding approach we use the developed simulation with a range of weeding scenarios captured from real fields. These models are captured using BonnBot-I from two crop fields (corn and sugar beet) observed over two years at Campus Klein-Altendorf (CKA) of the University of Bonn, as elaborated in Section 3.5.6. The field models used for training and evaluation were selected to represent all weed distributions without overlap between the subsets. This data reflects the key aspects relevant to testing our approach, including realistic field variations such as the performance under varying weed density, deployment for systems with multiple tools, and biodiversity-aware weeding. We compare the performance of the proposed RL method to the nOTSP algorithm as a baseline. The robot linear speed is set to $\gamma = 0.5m/s$ and the velocity of the linear actuators is $\vartheta = 5m/s$ to match the real robot operational conditions. We measure intervention loss as a percentage to indicate the number of weeds missed during weeding. Another important metric is the distance traveled by the weeding axes, in meters, which should ideally be low.

We trained the models using the PPO algorithm implemented in PyTorch with PyTorch Lightning on an Nvidia A4000 GPU. We used the Adam optimizer with an initial learning rate of $3 \exp -4$ and with a batch size of 64 to 256 and used an entropy coefficient of 0.01 to promote exploration. The training ran for $500 \exp 3$ epochs, with each epoch comprising multiple optimization steps every 50 epochs.

6.6.1 Single Axis Weeding Performance

To evaluate the performance of the proposed RL planning method, we directly compare it against the greedy planner and nOTSP (baseline) for conducting weeding on real-field models in simulation, demonstrated in Table 6.1. In our case, the greedy planner only tries approach to the closest target on the Y-axis. We use the RL planner with three different action spaces (constant, dynamic, and continuous). In this experiment, we consider that there is only one weeding tool. In scenarios with low weed density (CN20), the RL-continuous method outperforms its counterparts and achieves a loss (missed weeds) rate of 10.0% while only traveling $0.1m$ more than baseline (nOTSP). The advantage of the RL-continuous method becomes even more apparent in moderate to high weed densities (SB20-S1 and SB20-S2), where it not only maintains a substantial lead by halving the weeding loss to 24.9% and 35.9% with respect to the baseline but also maintains a low distance traveled. Moreover, in the most demanding scenario of very high weed density (SB21-S2), the RL-continuous approach continues to exhibit its superiority in both measures of weeding loss and deviled distance hence attributing smaller travel distance of the nOTSP w.r.t the RL method to the fact that it misses more weeds.

Table 6.1: Demonstrating the difference in the weeding performance of BonnBot-Iintervention pipeline using only a single linear axis with different control schemes: Greedy-planner, nOTSP (baseline), RL-constant, RL-dynamic, and RL-continuous. The metrics provide weeding loss (%) and traveled distance (m) of the linear axis. It is worth mentioning that the desired trend is a lower traveled distance with minimum missing weeds.

Fields			**Greedy-planner**		**nOTSP**		**RL-constant**		**RL-dynamic**		**RL-continuous**	
name	-	Avg/m^2	(%) ↓	(m)	(%) ↓	(m)	(%) ↓	(m)	(%) ↓	(m)	(%) ↓	(m)
CN20	low	3.1	22.0	16.9	19.5	17.1	82.7	42.4	74.6	41.1	**10.0**	**17.2**
SB20-S1	moderate	8.2	54.7	9.65	53.7	15.7	73.5	11.0	68.0	6.3	**24.9**	**17.8**
SB20-S2	high	15.4	75.5	16.2	72.8	15.3	33.9	10.1	82.6	9.4	**35.9**	**26.5**
SB21-S1	high	22.3	83.8	7.8	77.4	9.8	38.3	11.5	88.9	8.4	**19.6**	**18.6**
SB21-S2	very high	81.2	85.2	20.9	85.1	23.6	71.5	7.4	86.5	4.8	**64.9**	**26.6**

The advantage of the RL method in high-weed density scenarios can be attributed to the potential for a single spray to affect multiple weeds clustered closely together, which is reflected as multiple correct hits in our proposed reward function Equation (6.4). However, this effect leads to decreased accuracy in targeting plant centers, as anticipated.

To evaluate this, we conducted an experiment to compare the performance of RL models trained with and without the multi-hit factor incorporated into the reward function. Specifically, we trained another RL model without the multi-hit factor, so the agent could only get one weed sprayed with every single spray action. The results indicate that incorporating the multi-hit factor reduces the number of spray actions by an average of 5.3%, which positively impacts agrochemical savings. However, excluding the multi-hit factor from the reward function leads to an increase in the total loss of weeding performance, which is directly proportional to the weed density in the fields. For example with lower weed density, fewer instances of weeds clustered closely together with a distance less than spray footprints $\lambda = 0.02m$ will appear in the field models, hence fewer multi-hits could happen.

In the CN20 field model, which features the lowest weed density, the weeding loss of the models without multi-hit factor is %10.4, where its difference to the base models is negligible and only 0.4%. Conversely, in field models like SB21-S1 and SB21-S2, where dense clusters of weeds are more common, the exclusion of the multi-hit factor significantly decreases the weeding performance to 24.8% and 78.1%, respectively. This effect is also noticeable on SB20-S1 and SB20-S2 models with a moderate to high range of weed density with %27.4 and %41.1 weeding losses, respectively.

6.6.2 Effect of Varying the Observation Vector Length

We conducted an ablation study to evaluate the impact of observation vector length on our system's performance of RL agents. The observation vector length determines the number of closest targets (closest in the y-direction) that the RL agent can perceive simultaneously. We varied this length across different experiments, enabling agents to consider an observation vector of length ranging from 1 to 40 elements. The results, in Figure 6.5, show how the amount of information available to the RL agent influences its learning and decision-making process. A short observation vector length ($N = 1, 2$ or 4) restricts the agent's perception of the environment, potentially leading to sub-optimal decision-making. This could be due to the limited awareness of nearby targets and the ability to forecast upcoming situations. Conversely, a longer observation vector with length $N > 8$ gives the agent a more comprehensive view of the surroundings but has the downside of increased complexity leading to learning difficulties. We found empirically that $N = 5$ provided the best performance, so we use this setting in this chapter's other experiments.

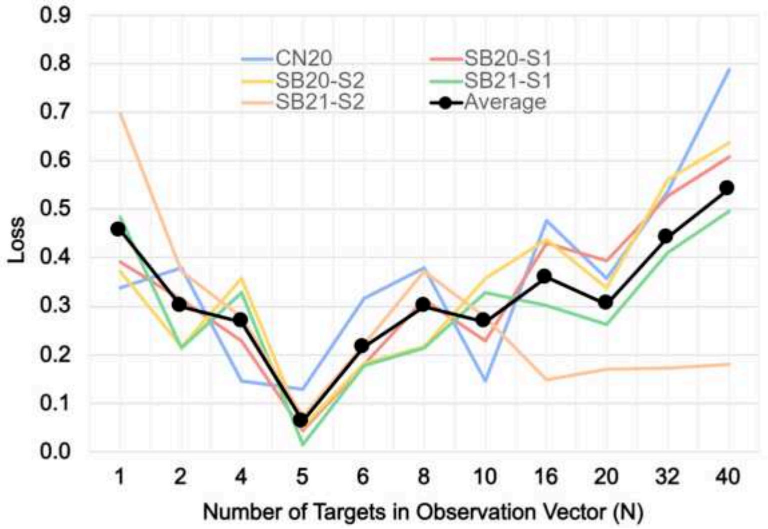


Figure 6.5: Plot to exploit the effect of observation vector length ($N = 1$ to $N = 40$) on the performance of weeding. The best model uses $N = 5$ which on average leads to the least weeding loss.

Table 6.2: Weeding performances of BonnBot-Iweeding tool using nOTSP (baseline) and RL-based models (RL-continuous).

Fields			nOTSP		RL-based	
	-	Avg/m^2	(%)↓	(m)	(%)↓	(m)
CN20	low	3.1	**0.0**	2.0±1.6	6.9	4.8±3.2
SB20-S1	moderate	8.2	**0.0**	4.1±2.1	4.3	3.4±0.9
SB20-S2	high	15.4	6.4	1.9±0.9	**5.1**	6.2±1.9
SB21-S1	high	22.3	14.1	2.8±2.6	**1.5**	4.5±3.3
SB21-S2	very high	81.2	36.5	5.2±3.1	**7.2**	7.3±3.4

6.6.3 Multi-Axis Weeding Performance

In this experiment, we provide a comparative analysis of a multi-axis weeding performance. We use two planning models: nOTSP and RL, where under a new experimental setup, BonnBot-I utilizes 4 linear axes with the static-subdivision method for target management within the workspace, where each linear axis has an independent weeding planner (nOTSP or RL). The static sub-division method divides the total width of the weeding tool's workspace into sections based on the number of available weeding heads. Consequently, each weeding head is assigned to a smaller portion of the workspace (see Section 5.4). Similar to our single axes evaluations Equation (6.6.1), in Table 6.2 we examine weed densities ranging from low to very high.

In Table 6.2, the low and moderate weed-density scenarios (CN20 and SB20-S1), the nOTSP method achieves a remarkable weed management efficiency, indicated by a 0.0% weeding loss. This suggests that, for lower densities, the nOTSP is highly effective, potentially due to its ability to better navigate and identify weeds within less cluttered environments. By contrast, the RL-based system misses some weeds and has a weeding loss of 6.9% and 4.3%, respectively.

As weed density increases to high and very high levels (SB20-S2, SB21-S1, SB21-S2), the RL-based method begins to outperform the nOTSP in managing weeds while also keeping the traveled distance relatively low. For instance, in the SB20-S2 scenario, the RL-based method reduces the weeding loss to 5.1% compared to 6.4% by the nOTSP, albeit at a higher distance travelled. This trend continues in more densely weeded environments (SB21-S1 and SB21-S2), with the RL-based method significantly reducing weeding loss compared to the nOTSP, suggesting that the RL-based approach can handle more complex densely weeded scenarios.

The RL-based system shows increased travel distances in high weed-density areas, indicating a trade-off between weed reduction and movement efficiency. This highlights the potential of RL-based systems to adapt to complex environments. While nOTSP is efficient in low weed densities, the RL approach excels in dense weed areas despite longer distances traveled.

6.6.4 Biodiversity-Aware Weeding

Our previous evaluations only considered how often we missed weeds in each crop row and did not consider the importance of each weed. For precision weeding, this is not the optimal solution as some weeds do not require intervention as they have minimal impact on the crop. In this evaluation, we consider weeding decisions based on biodiversity-aware schemes. Moreover, we double the system's speed to $\gamma = 1m/s$ to represent a stressed condition for weeding. Hence, this experiment shows the effectiveness of learned policy compared to the biodiversity-aware variant of nOTSP. In this experiment, planning models use 4 linear axes as in the previous section.

In Table 6.3, we compare three approaches: biodiversity-aware nOTSP (nOTSP-Bio-Div), RL-based, and a special variant of the RL-based model, guided by our novel reward function to consider biodiversity (RL-Bio-Div), as introduced in Section 6.5. To facilitate the analysis of experimental results, we combined the weeding loss percentage of low and high-priority weeds with $\alpha \times w_h + (1 - \alpha) \times w_l$ where w_h and w_l denote high-priority and low-priority loss rates, respectively. We considered the field models to only contain two classes of low and high-priority weeds (with a ratio of 1 to 10 both in count and priority), hence, setting $\alpha = 0.9$. Generally, RL-Bio-Div shows promising results across all scenarios, lowering the weeding loss to an average of 26.7%. Notably, in SB20-S1, RL-Bio-Div achieves a loss rate of 14.2%, outperforming both RL-based and nOTSP-Bio-Div with 27.2%

and 5.2%, respectively.

Table 6.3: Weeding loss of nOTSP-Bio-Div, RL-based and RL-Bio-Div methods on real-field models.

Fields	**nOTSP-Bio-Div** (%)↓	**RL-based** (%)↓	**RL-Bio-Div** (%)↓
CN20	41.8	26.3	**24.7**
SB20-S1	41.4	19.4	**14.2**
SB20-S2	38.8	**29.7**	30.9
SB21-S1	33.6	29.9	**28.3**
SB21-S2	79.3	41.9	**35.5**

The RL-Bio-Div performance improvements are particularly pronounced in high weed-density scenarios, such as SB21-S2, where it outperforms the nOTSP-Bio-Div relatively by 43.8% and the RL-based by 6.4%. These results highlight the importance of considering weed density variations when evaluating weeding strategies and emphasize the potential of RL-Bio-Div in optimizing weeding performance across diverse field conditions.

6.7 Conclusion

In this chapter, we significantly advanced the weeding capabilities of BonnBot-I by introducing a novel and reactive reinforcement learning (RL)-based scheme. The proposed method enables precision weeding utilizing kinematics and observation models of our self-built weeding robot, BonnBot-I. This development is closely tied to the last research question posed in Chapter 1: "*Can a robotic system learn to conduct precision weed management based on diverse plant characteristics and priorities provided by experts?*" Our approach leverages the power of RL to optimize the movement strategies of independently controllable weeding nozzles, allowing for flexible and efficient operations based on multi-modal observations.

The RL algorithm we employed is derived from the proximal policy optimization (PPO) method, a state-of-the-art technique known for its stability and efficiency in training complex policies. By modeling the precision weeding task as an RL problem, we enabled BonnBot-I to learn and adapt its intervention strategies dynamically. This learning process accounts for the robot's real kinematics and the partial observability of the environment, making it well-suited for the highly variable and dynamic conditions of real-world agricultural fields.

One of the critical advantages of using RL in this context is its ability to continuously refine its decision-making process through interaction with the environment. This is particularly beneficial in precision agriculture, where the variability in weed density, crop types, and environmental conditions requires a system that can adapt on-the-fly. Our RL-based approach allows BonnBot-I to integrate multiple sources of informationsuch as visual data from cameras, spatial data from sensors, and expert-provided plant prioritiesto make informed decisions about weeding actions.

In Section 6.6, we present the results of extensive experiments conducted using real-world field models of sugar beet and corn crops. These experiments demonstrate the effectiveness of our RL-based method, which reduced the number of untreated weeds by an average of 5.0% compared to state-of-the-art traveling salesman problem (TSP)-based planners. Furthermore, our innovative biodiversity-aware RL approach improved weed treatment in complex and diverse field conditions by 26.26% over the nOTSP-based biodiversity-aware method. This improvement is a testament to the system's ability to handle the complexity and diversity of real-world agricultural environments, making it a significant step forward in the field of precision farming.

We attribute the success of our approach to several key factors. First, the RL algorithm's capability to perform high-frequency re-planning allows it to respond effectively to the partial observability of crop rows, a common challenge in agricultural robotics. Second, the decentralized policy execution within the RL framework enables smooth transitions between single-agent and multi-agent operations, providing a versatile solution that can scale to different field conditions and operational requirements.

This research lays a strong foundation for future work in the field of robotic precision agriculture. One promising direction is to enhance the efficiency of the RL-based system by further optimizing its movement strategies to reduce travel distances while maintaining high weed removal efficacy. Another critical area for future exploration is the deployment of these algorithms in real-world applications, which presents significant challenges due to the sim-to-real gap. Bridging this gap will require more sophisticated learning algorithms, improved simulation environments, and potentially, the refinement of target management strategies, such as the static-subdivision method, to ensure that the system performs reliably in diverse agricultural settings. In conclusion, this chapter not only demonstrates the potential of RL to transform precision weeding but also opens up new path for research and development in the broader context of agricultural robotics.

Chapter 7

Conclusion

This thesis investigated the application of robotic vision and automation in precision weed management and crop monitoring within arable farming. We presented research topics focusing on developing advanced techniques that combine classical computer vision and deep learning to enhance the autonomy, accuracy, and effectiveness of agricultural robots. By integrating these technologies, we demonstrated how robotic systems can significantly improve field navigation, weed control, and sustainable crop management, advancing the state-of-the-art in precision agriculture. We showcased the BonnBot-Is ability to execute targeted, biodiversity-aware weeding, emphasizing the importance of plant-level precision to reduce herbicide use and promote ecological balance. We structured the thesis into six chapters, each focusing on different fundamental research questions introduced in Section 1 ranging from robot design to autonomous precision agriculture, aspects of robotic vision in agricultural sites, bio-diversity-aware precision weeding:

The foundation of this work is built on BonnBot-I introduced in Chapter 3, our novel precision crop and weed management robot, specifically designed to address the complexities of plant-level interventions in dynamic agricultural environments. We introduced the BonnBot-I advanced sensor systems and intervention tools that are engineered to operate reliably in real-world conditions, autonomously navigating fields and performing precise weeding actions that prioritize environmental sustainability. The chapter also covers the simulation environments developed specifically for BonnBot-I facilitating extensive testing and training of weeding algorithms under controlled conditions. Furthermore, we explained our DNN-based monitoring approach which was developed to offer crop- and platform-agnostic monitoring operations in different environments. Moreover, it highlights the three unique, publicly available datasets collected over three years at the University of Bonn's Campus Klein-Altendorf, used to validate the methods developed throughout this research.

In Chapter 4, we focused on autonomy in agricultural fields, showcasing vision-based navigation strategies that enable BonnBot-Is to navigate row-crop fields using only on-board cameras. This chapter discusses the core principles of autonomous navigation, emphasizing the challenges faced in dynamic, cluttered environments. Our method integrated multi-crop-row switching to its control scheme, ensuring the whole field can be traversed. The experiments show that the presented approach can reliably detect multi-crop-row structures of various row-crop fields, guide the robot along the lanes of crops in five different fields accurately and, autonomously switch between adjacent lanes of crops without supervision successfully replacing manual operation.

In Chapter 5 delves into the key question of plant-level, biodiversity-aware weeding using robotic vision. This chapter reviewed related robotic weeding approaches and outlined the system requirements for implementing precision weeding with BonnBot-I. It introduced a method for modeling real-field observations into usable formats, crucial for accurate weeding actions. The chapter further discussed target-space management strategies and path-planning algorithms that control BonnBot-Is intervention tools, culminating in a biodiversity-aware weeding scheme that prioritizes ecologically sensitive weeding actions. Experimental evaluations in both simulated and real-world conditions demonstrate the system's performance in sugar beet and corn fields, highlighting its effectiveness in sustainable weed management.

Finally, the application of reinforcement learning (RL) is investigated for the first time in Chapter 6 to optimize plant-level interventions. We introduced an RL framework developed for BonnBot-I, designed to enhance the BonnBot-I's ability to learn and adapt its weeding strategies based on real-time observations. The chapter covers the design of the observation models, action spaces, and reward functions that underpin the RL-based approach, enabling the robot to make intelligent, context-aware weeding decisions. Experimental results demonstrate the advantages of the RL-driven system over traditional methods, showing improved intervention accuracy and efficiency.

Overall, this thesis illustrated significant contributions of the pursued research topic to showcase further the potential of combining robotic vision and advanced learning techniques to enhance the autonomy, precision, and sustainability of agricultural robots. By implementing these technologies on BonnBot-I, we demonstrate significant improvements in the accuracy and effectiveness of field interventions, redirecting the precision weeding to a new era of sustainable, tech-driven agriculture. We can confidently assert that BonnBot-I has paved the way for the future of biodiversity-aware, plant-level weeding and interventions in agriculture. Its innovative capabilities have transformed how autonomous robots handle targeted treatments, ensuring precision and minimal environmental impact. Looking ahead, BonnBot-I is positioned to considerably advance field management techniques considerably, supporting more sustainable and eco-conscious farming practices that promote healthier ecosystems while optimizing crop production.

7.1 Open Source Contributions

During this project, we made several open-source contributions which are listed in below:

Datasets

- 2020 - Sugar-beet Dataset SB20 `https://agrobotics.uni-bonn.de/sugar_beet_2020_dataset/index.html`.
- 2021 - Corn Dataset CN20 `https://agrobotics.uni-bonn.de/corn_2020_dataset/index.html`.
- 2021 - Multi-Crop Row Navigation Dataset MultiNav21`http://github.com/PRBonn/visual-crop-row-navigation`.
- 2023 - Sugar-beet Dataset SB21 `https://agrobotics.uni-bonn.de/sugar_beet_2021_dataset/index.html`.

Software Packages

- 2021 - Multi-Crop Row Navigation Package `http://github.com/PRBonn/visual-crop-row-navigation`.
- 2024 - RL-Gym environment along with Weeding Specific Field models WeedingGym-v1`https://agrobotics.uni-bonn.de/rl_weeding_gym/index.html`.

7.2 Future Work

The advancements presented in this thesis highlight the potential of integrating robotic vision, machine learning, and autonomous intervention strategies in precision agriculture. While significant progress has been made, numerous opportunities for future work could further enhance the capabilities, efficiency, and sustainability of agricultural robots like BonnBot-I. The following sections outline key areas for future research and development, emphasizing the expansion of precision weeding, the integration of advanced perception technologies, the exploration of multi-modal data fusion, and the enhancement of autonomous navigation and decision-making systems.

Advanced Perception and Multi-Modal Data Integration A critical area for future work could be the enhancement of perception systems through the integration of multi-modal data sources, including LiDAR, thermal cameras, and hyperspectral sensors, and further advancing machine vision models for higher levels of adaptability. Currently, BonnBot-Irelies primarily on RGB-D and NIR cameras for field navigation and target detection, which, while effective, can be limited under challenging conditions such as variable lighting, shadows, or high weed density. Incorporating additional sensors can considerably improve the robots ability to distinguish between crops and weeds under diverse conditions, enhancing detection accuracy and robustness.

Further exploration into multi-modal data fusion techniques can enable the development of more sophisticated perception models that leverage complementary information from different sensors, leading to more accurate and reliable plant-level identification. Future research should focus on creating unified data processing pipelines that integrate these varied data sources, allowing the robot to make more informed decisions based on a comprehensive understanding of its environment.

Enhanced Decision-Making with RL While the reinforcement learning (RL) framework implemented in this thesis has demonstrated promising results in optimizing plant-level interventions, there is significant potential to expand these capabilities. Future work could explore more advanced RL algorithms, such as hierarchical reinforcement learning or meta-learning, which allow the robot to learn from past experiences more efficiently and adapt its strategies based on real-time changes in the field environment. This approach would enable BonnBot-Ito dynamically adjust its weeding strategies according to evolving field conditions, such as weed density fluctuations, crop growth stages, or unexpected obstacles.

Moreover, incorporating adaptive control techniques that enable the robot to continuously refine its actions based on sensor feedback could further improve intervention precision. This would involve developing adaptive models that can adjust intervention parameters, such as spray rates or tool positions, controlling the type of intervention, in response to real-time data, and ensuring optimal performance in varying field conditions.

Scalability and Real-World Deployment The current implementation of BonnBot has been extensively tested in controlled environments and specific crop settings, such as sugar beet and corn fields. However, scalability to a broader range of crops and field conditions remains a critical challenge. Future work should aim to validate and optimize the system across diverse cropping systems, including row crops, orchards, and mixed planting scenarios, each presenting unique challenges for navigation and intervention.

To facilitate real-world deployment, further research is needed to address the operational scalability of BonnBot-I, focusing on energy efficiency, system robustness, and maintenance requirements. Developing lightweight and energy-efficient hardware components, alongside more durable and weather-resistant designs, would ensure the robot can operate reliably over extended periods and larger field areas. Additionally, integrating cloud-based data management and remote control systems could enable real-time monitoring and adjustments, enhancing the robots usability in commercial farming operations.

Improved Biodiversity-Aware Weeding Strategies Bio-diversity-aware weeding has the potential to revolutionize sustainable farming practices by reducing reliance on chemical herbicides and promoting ecological balance. The missing part of the puzzle at the moment is a proper model of environmental impact from diverse constlaion of weeds and crops. Developing and utilizing such assessment models in the decision-making framework could enable the robot to optimize interventions based on immediate crop health and long-term ecosystem health.

Collaborative Robotics and Integrated Farm Management Scaling precision weeding across larger, diverse fields can be optimized through collaborative multi-robot systems, where robots coordinate tasks and share data for greater efficiency. By leveraging swarm robotics principles, these systems enhance coverage and resilience, distributing tasks among specialized robots for interventions like spot spraying and soil sampling. Developing protocols for task coordination would support autonomous large-scale operations. Integrating robots with farm management systems enables seamless data exchange, empowering farmers with real-time insights for data-driven decisions in irrigation, nutrient application, and yield prediction. This ultimately reduces input costs and enhances productivity.

Overall we believe, the future of precision agriculture lies in the continued development and integration of advanced robotic vision, machine learning, and autonomous intervention technologies. By expanding the capabilities of agricultural robots, and enhancing their perception, decision-making, and collaborative functions, we can drive the next wave of innovations that will make farming more sustainable, efficient, and resilient.

Bibliography

[1] Garford(2014).robocrop inrow weeder. `https://garford.com/`. Accessed: 2020-10-22.

[2] ABARES. Farm financial performance. In *Australian Bureau of Agriculture and Resource Economics and Sciences (ABARES)*, 2018.

[3] Alireza Ahmadi, Michael Halstead, and Chris McCool. Towards autonomous crop-agnostic visual navigation in arable fields. *arXiv preprint arXiv:2109.11936*, 2021.

[4] Alireza Ahmadi, Michael Halstead, and Chris McCool. Virtual temporal samples for recurrent neural networks: applied to semantic segmentation in agriculture. In *German conference on pattern recognition*, 2021.

[5] Alireza Ahmadi, Michael Halstead, and Chris McCool. Bonnbot-i: A precise weed management and crop monitoring platform. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9202–9209. IEEE, 2022.

[6] Alireza Ahmadi, Michael Halstead, Claus Smitt, and Chris McCool. Bonnbot-i plus: A bio-diversity aware precise weed management robotic platform. *arXiv e-prints*, pages arXiv–2405, 2024.

[7] Alireza Ahmadi, Lorenzo Nardi, Nived Chebrolu, and Cyrill Stachniss. Visual servoing-based navigation for monitoring row-crop fields. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4920–4926. IEEE, 2020.

[8] Amazone. `https://amazone.net/`, 2024.

[9] Sandra Amend, David Brandt, Daniel Di Marco, Tobias Dipper, Gabriel Gässler, Markus Höferlin, Maurice Gohlke, Katharina Kesenheimer, Peter Lindner, Roland Leidenfrost, et al. Weed management of the future. *KI-Künstliche Intelligenz*, 33(4):411–415, 2019.

[10] J Ascard, PE Hatcher, B Melander, MK Upadhyaya, and RE Blackshaw. 10 thermal weed control. *Non-chemical weed management: principles, concepts and technology*, pages 155–175, 2007.

[11] B. Åstrand and A. J. Baerveldt. A vision based row-following system for agricultural field machinery. *Mechatronics*, 15(2):251–269, 2005.

[12] Björn Åstrand and Albert-Jan Baerveldt. A vision based row-following system for agricultural field machinery. *Mechatronics*, 15(2):251–269, 2005.

[13] Marianne Bakken, Richard JD Moore, and Pål From. End-to-end learning for autonomous crop row-following. *IFAC-PapersOnLine*, 52(30):102–107, 2019.

[14] Marianne Bakken, Vignesh Raja Ponnambalam, Richard JD Moore, Jon Glenn Omholt Gjevestad, and Pål Johan From. Robot-supervised learning of crop row segmentation.

[15] Tijmen Bakker, Kees van Asselt, Jan Bontsema, Joachim Müller, and Gerrit van Straten. An autonomous weeding robot for organic farming. In *Field and Service Robotics*, pages 579–590. Springer, 2006.

[16] Tijmen Bakker, Jan Bontsema, Joachim Müller, et al. Systematic design of an autonomous platform for robotic weeding. *Journal of Terramechanics*, 47(2):63–73, 2010.

[17] Oscar C Barawid Jr, Akira Mizushima, Kazunobu Ishii, and Noboru Noguchi. Development of an autonomous navigation system using a two-dimensional laser scanner in an orchard application. *Biosystems Engineering*, 96(2):139–149, 2007.

[18] Owen Bawden, Jason Kulk, Ray Russell, Chris McCool, Andrew English, Feras Dayoub, Chris Lehnert, and Tristan Perez. Robot for weed species plant-specific management. *Journal of Field Robotics*, 34(6):1179–1199, 2017.

[19] Owen Bawden, Jason Kulk, Ray Russell, Chris McCool, Andrew English, Feras Dayoub, Chris Lehnert, and Tristan Perez. Robot for weed species plantspecific management. *Journal of Field Robotics*, 34:1179–1199, 2017.

[20] M Bayati and R Fotouhi. A mobile robotic platform for crop monitoring. *Adv. Robot. Autom*, 7(2), 2018.

[21] J Billingsley and M Schoenfisch. The successful development of a vision guidance system for agriculture. *Computers and electronics in agriculture*, 16(2):147–163, 1997.

[22] Cian Blaix, Anna Camilla Moonen, DF Dostatny, J Izquierdo, Josiane Le Corff, J Morrison, C Von Redwitz, M Schumacher, and PR Westerman. Quantification of regulating ecosystem services provided by weeds in annual cropping systems using a systematic map approach. *Weed research*, 58(3):151–164, 2018.

[23] José Blasco, Nuria Aleixos, JM Roger, Gilles Rabatel, and Enrique Moltó. Aeautomation and emerging technologies: Robotic weed control using machine vision. *Biosystems Engineering*, 83(2):149–157, 2002.

[24] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9157–9166, 2019.

[25] Stephanie Bonadies and S Andrew Gadsden. An overview of autonomous crop row navigation strategies for unmanned ground vehicles. *Engineering in Agriculture, Environment and Food*, 12(1):24–31, 2019.

[26] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[27] Chung-Liang Chang, Bo-Xuan Xie, and Sheng-Cheng Chung. Mechanical control with a deep learning method for precise weeding on a farm. *Agriculture*, 11(11):1049, 2021.

[28] Chung-Liang Chang, Bo-Xuan Xie, and Sheng-Cheng Chung. Mechanical control with a deep learning method for precise weeding on a farm. *Agriculture*, 11(11), 2021.

[29] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[30] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12475–12485, 2020.

[31] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image

segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022.

[32] A. Cherubini, F. Chaumette, and G. Oriolo. An image-based visual servoing scheme for following paths with nonholonomic mobile robots. In *Proc. of the Intl. Conf. on Control, Automation, Robotics and Vision*, pages 108–113, 2008.

[33] Hock Hung Chieng and Noorhaniza Wahid. A performance comparison of genetic algorithms mutation operators in n-cities open loop travelling salesman problem. In *Recent Advances on Soft Computing and Data Mining*, pages 89–97. Springer, 2014.

[34] Danilo Alves de Lima and Alessandro Corrêa Victorino. A visual servoing approach for road lane following with obstacle avoidance. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 412–417. IEEE, 2014.

[35] Rajitha de Silva, Grzegorz Cielniak, and Junfeng Gao. Towards agricultural autonomy: crop row detection under varying field conditions using deep learning. *arXiv preprint arXiv:2109.08247*, 2021.

[36] John Deere. `https://www.deere.com/en/sprayers/see-spray-ultimate/`, 2024.

[37] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, et al. Robothor: An open simulation-to-real embodied ai platform. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3164–3174, 2020.

[38] Pauline Dentika, Harry Ozier-Lafontaine, and Laurent Penet. Weeds as pathogen hosts and disease risk for crops in the wake of a reduced use of herbicides: Evidence from yam (dioscorea alata) fields and colletotrichum pathogens in the tropics. *Journal of Fungi*, 7(4):283, 2021.

[39] Earthrover. `https://www.earthrover.farm/`, 2024.

[40] Bernard Espiau, François Chaumette, and Patrick Rives. A new approach to visual servoing in robotics. *ieee Transactions on Robotics and Automation*, 8(3):313–326, 1992.

[41] Felix Esser, Elias Marks, Federico Magistri, Jan Weyler, Simon Bultmann, Tobias Zaenker, Alireza Ahmadi, Michael Schreiber, Heiner Kuhlmann,

Chris McCool, et al. Automated leaf-level inspection of crops in agricultural fields by combining aerial and ground robot systems.

[42] Felix Esser, Radu Alexandru Rosu, André Corneließen, Lasse Klingbeil, Heiner Kuhlmann, and Sven Behnke. Field robot for high-throughput and high-resolution 3d plant phenotyping: towards efficient and sustainable crop production. *IEEE Robotics & Automation Magazine*, 2023.

[43] Xiangpeng Fan, Xiujuan Chai, Jianping Zhou, and Tan Sun. Deep learning based weed detection and target spraying robot system at seedling stage of cotton field. *Computers and Electronics in Agriculture*, 214:108317, 2023.

[44] Farmdroid. `https://farmdroid.com/`, 2024.

[45] Mulham Fawakherji, Ali Youssef, Domenico Bloisi, Alberto Pretto, and Daniele Nardi. Crop and weeds classification for precision agriculture using context-independent pixel-wise segmentation. In *2019 third IEEE international conference on robotic computing (IRC)*, pages 146–152. IEEE, 2019.

[46] Olga Fishkis and Heinz-Josef Koch. Effect of mechanical weeding on soil erosion and earthworm abundance in sugar beet (beta vulgaris l.). *Soil and Tillage Research*, 225:105548, 2023.

[47] Gazebo. `https://classic.gazebosim.org/`, 2024.

[48] Lars Grimstad and Pål J From. Software components of the thorvald ii modular robot. 2018.

[49] Lars Grimstad and Pål Johan From. Thorvald ii-a modular and reconfigurable agricultural robot. *IFAC-PapersOnLine*, 50(1):4588–4593, 2017.

[50] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[51] M. Halstead, S. Denman, C. Fookes, and C. McCool. Fruit detection in the wild: The impact of varying conditions and cultivar. In *Proceedings of Digital Image Computing: Techniques and Applications (DICTA)*, 2020.

[52] Michael Halstead, Alireza Ahmadi, Claus Smitt, Oliver Schmittmann, and Chris Mccool. Crop agnostic monitoring driven by deep learning. *Frontiers in Plant Science*, page 2937, 2021.

[53] Michael Halstead, Christopher McCool, Simon Denman, Tristan Perez, and Clinton Fookes. Fruit quantity and ripeness estimation using a robotic vision system. *IEEE Robotics and Automation Letters*, 3(4):2995–3002, 2018.

[54] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[55] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[56] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.

[57] Elia Kaufmann, Leonard Bauersfeld, Antonio Loquercio, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Champion-level drone racing using deep reinforcement learning. *Nature*, 620(7976):982–987, 2023.

[58] Florian Kraemer, Alexander Schaefer, Andreas Eitel, Johan Vertens, and Wolfram Burgard. From plants to landmarks: Time-invariant plant localization that uses deep pose regression in agricultural fields. *arXiv preprint arXiv:1709.04751*, 2017.

[59] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[60] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[61] James Ju Heon Lee, Kris Frey, Robert Fitch, and Salah Sukkarieh. Fast path planning for precision weeding. In *Australasian Conference on Robotics and Automation, ACRA*, 2014.

[62] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[63] Yong Li, Zhiqiang Guo, Feng Shuang, Man Zhang, and Xiuhua Li. Key technologies of machine vision for weeding robots: A review and benchmark. *Computers and Electronics in Agriculture*, 196:106880, 2022.

[64] Yu-Kai Lin and Shih-Fang Chen. Development of navigation system for tea field machine using semantic segmentation. *IFAC-PapersOnLine*, 52(30):108–113, 2019.

[65] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[66] Philipp Lottes, Jens Behley, Andres Milioto, and Cyrill Stachniss. Fully convolutional networks with sequential information for robust crop and weed detection in precision farming. *IEEE Robotics and Automation Letters*, 3(4):2870–2877, 2018.

[67] JF MacGregor and TJ Harris. The exponentially weighted moving variance. *Journal of Quality Technology*, 25(2):106–118, 1993.

[68] Flavio BP Malavazi, Remy Guyonneau, Jean-Baptiste Fasquel, Sebastien Lagrange, and Franck Mercier. Lidar-only based navigation algorithm for an autonomous agricultural robot. *Computers and electronics in agriculture*, 154:71–79, 2018.

[69] Anne Maredia and R Pepper. History, analysis, and implementation of traveling salesman problem (tsp) and related problems. *Department of Computer and Mathematical Sciences, University of Houston-Downtown*, 2010.

[70] Anna Massfeller, Marie Zingsheim, Alireza Ahmadi, Elin Martinsson, and Hugo Storm. Action-or results-based payments for ecosystem services in the era of smart weeding robots? *Biological Conservation*, 302(11099):8, 2025.

[71] Rajesh Matai, Surya Prakash Singh, and Murari Lal Mittal. Traveling salesman problem: an overview of applications, formulations, and solution approaches. *Traveling salesman problem, theory and applications*, 1(1):1–25, 2010.

[72] Chris McCool, James Beattie, Jennifer Firn, Chris Lehnert, Jason Kulk, Owen Bawden, Raymond Russell, and Tristan Perez. Efficacy of mechanical weeding tools: A study into alternative weed management strategies enabled by robotics. *IEEE Robotics and Automation Letters*, 3(2):1184–1190, 2018.

[73] Chris McCool, Tristan Perez, and Ben Upcroft. Mixtures of lightweight deep convolutional neural networks: Applied to agricultural robotics. *IEEE Robotics and Automation Letters*, 2(3):1344–1351, 2017.

[74] Md Jebu Mia, Francesca Massetani, Giorgio Murri, and Davide Neri. Sustainable alternatives to chemicals for weed control in the orchard–a review. *Horticultural Science*, 47(1):1–12, 2020.

[75] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[76] OpenAI. `https://openai.com/index/openai-gym-beta/`, 2024.

[77] Manuel Pérez-Ruíz, David C Slaughter, Fadi A Fathallah, Chris J Gliever, and Brandon J Miller. Co-robotic intra-row weed control system. *Biosystems engineering*, 126:45–55, 2014.

[78] Andrea Peruzzi, Luisa Martelloni, Christian Frasconi, Marco Fontanelli, Michel Pirchio, and Michele Raffaelli. Machines for non-chemical intra-row weed control in narrow and wide-row crops: a review. *Journal of Agricultural Engineering*, 48(2):57–70, 2017.

[79] Vignesh Raja Ponnambalam, Marianne Bakken, Richard JD Moore, Jon Glenn Omholt Gjevestad, and Pål Johan From. Autonomous crop row guidance using adaptive multi-roi in strawberry fields. *Sensors*, 20(18):5249, 2020.

[80] Alberto Pretto, Stéphanie Aravecchia, Wolfram Burgard, Nived Chebrolu, Christian Dornhege, Tillmann Falck, Freya Veronika Fleckenstein, Alessandra Fontenla, Marco Imperoli, Raghav Khanna, et al. Building an aerial–ground robotics system for precision farming: An adaptable solution. *IEEE Robotics & Automation Magazine*, 28(3):29–49, 2020.

[81] Manuel Pérez-Ruíz, David C. Slaughter, Fadi A. Fathallah, Chris J. Gliever, and Brandon J. Miller. Co-robotic intra-row weed control system. *Biosystems Engineering*, 126:45–55, 2014.

[82] Peter H Raven and David L Wagner. Agricultural intensification and climate change are rapidly decreasing insect biodiversity. *Proceedings of the National Academy of Sciences*, 118(2), 2021.

[83] J Redmon. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[84] Clemens Reimann, Peter Filzmoser, and Robert G Garrett. Background and threshold: critical comparison of methods of determination. *Science of the total environment*, 346(1-3):1–16, 2005.

[85] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[86] Farming Revolution. `https://farming-revolution.com/`, 2024.

[87] Carbon Robotics. `https://carbonrobotics.com/`, 2024.

[88] Deepfield Robotics. Bonirob: adaptable multi-purpose robotic platform, 2015.

[89] Eco Robotics. `https://ecorobotix.com/`, 2024.

[90] Pixel Farming Robotics. `https://pixelfarmingrobotics.com/`, 2024.

[91] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[92] ROS. `https://ros.org/`, 2024.

[93] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019.

[94] SBG Systems. Technical Data Ellipse2-D, 2020. `https://www.sbg-systems.com/products/ellipse-series/#ellipse-d_rtk_gnss_ins`.

[95] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[96] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[97] DC Slaughter, DK Giles, and D Downey. Autonomous robotic weed control systems: A review. *Computers and electronics in agriculture*, 61(1):63–78, 2008.

[98] D.C. Slaughter, D.K. Giles, and D. Downey. Autonomous robotic weed control systems: A review. *Computers and Electronics in Agriculture*, 61(1):63–78, 2008. Emerging Technologies For Real-time and Integrated Agriculture Decisions.

[99] Claus Smitt, Michael Halstead, Alireza Ahmadi, and Chris McCool. Explicitly incorporating spatial information to recurrent networks for agriculture. *IEEE Robotics and Automation Letters*, 2022.

[100] Claus Smitt, Michael Halstead, Tobias Zaenker, Maren Bennewitz, and Chris McCool. Pathobot: A robot for glasshouse crop phenotyping and intervention. In *2021 International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[101] Claus Smitt, Michael Halstead, Tobias Zaenker, Maren Bennewitz, and Chris McCool. Pathobot: A robot for glasshouse crop phenotyping and intervention. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2324–2330. IEEE, 2021.

[102] Henning Tangen Søgaard and Hans Jørgen Olsen. Determination of crop rows by image analysis without segmentation. *Computers and electronics in agriculture*, 38(2):141–158, 2003.

[103] Yunlong Song, Angel Romero, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Reaching the limit in autonomous racing: Optimal control versus reinforcement learning. *Science Robotics*, 8(82):eadg1462, 2023.

[104] Brian L Steward, Lei F Tian, and Lie Tang. Distance–based control system for machine vision–based selective spraying. *Transactions of the ASAE*, 45(5):1255, 2002.

[105] Peter Sturm. Pinhole camera model. In *Computer Vision: A Reference Guide*, pages 983–986. Springer, 2021.

[106] Ling Tan, Lei Sun, Boyuan Cao, Jingming Xia, and Hai Xu. Research on weighted energy consumption and delay optimization algorithm based on dual-queue model. *IET Communications*, 18(1):81–95, 2024.

[107] Escarda Technologies. `https://www.escarda.tech/`, 2024.

[108] Naio Technologies. `https://www.naio-technologies.com/`, 2024.

[109] Benoît Thuilot, Christophe Cariou, Philippe Martinet, and Michel Berducat. Automatic guidance of a farm tractor relying on a single cp-dgps. *Autonomous robots*, 13(1):53–71, 2002.

[110] Yunong Tian, Guodong Yang, Zhe Wang, Hao Wang, En Li, and Zize Liang. Apple detection during different growth stages in orchards using the improved yolo-v3 model. *Computers and electronics in agriculture*, 157:417–426, 2019.

[111] Nihan Sölpük Turhan. Karl pearson's chi-square tests. *Educational Research and Reviews*, 16(9):575–580, 2020.

[112] Trygve Utstumo, Frode Urdal, Anders Brevik, Jarle Dørum, Jan Netland, Øyvind Overskeid, Therese W Berge, and Jan Tommy Gravdahl. Robotic in-row weed control in vegetables. *Computers and electronics in agriculture*, 154:36–45, 2018.

[113] Eldert J Van Henten, Jochen Hemming, BAJ Van Tuijl, JG Kornet, J Meuleman, Jan Bontsema, and EA Van Os. An autonomous robot for harvesting cucumbers in greenhouses. *Autonomous robots*, 13(3):241–258, 2002.

[114] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[115] Weedbot. `https://weedbot.eu/`, 2024.

[116] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. *arXiv preprint arXiv:1911.00357*, 2019.

[117] Wera Winterhalter, Freya Veronika Fleckenstein, Christian Dornhege, and Wolfram Burgard. Crop row detection on tiny plants with the pattern hough transform. *IEEE Robotics and Automation Letters*, 3(4):3394–3401, 2018.

[118] D. Woebbecke, G. Meyer, K. Von Bargen, and D. A. Mortensen. Color indices for weed identification under various soil, residue, and lighting conditions. *Trans. of the American Society of Agricultural Engineers (ASAE)*, 38(1):259–269, 1995.

[119] Xiaolong Wu, Stéphanie Aravecchia, Philipp Lottes, Cyrill Stachniss, and Cédric Pradalier. Robotic weed control using automated weed and crop classification. *Journal of Field Robotics*, 37(2):322–340, 2020.

[120] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

[121] Ya Xiong, Yuanyue Ge, Yunlin Liang, and Simon Blackmore. Development of a prototype robot and fast path-planning algorithm for static laser weeding. *Computers and Electronics in Agriculture*, 142:494–503, 2017.

[122] Ilaria Zambon, Massimo Cecchini, Gianluca Egidi, Maria Grazia Saporito, and Andrea Colantoni. Revolution 4.0: Industry vs. agriculture in a future development for smes. *Processes*, 7(1):36, 2019.

[123] Mingchuan Zhou, Huanyu Jiang, Zhenshan Bing, Hang Su, and Alois Knoll. Design and evaluation of the target spray platform. *International Journal of Advanced Robotic Systems*, 18(2):1729881421996146, 2021.

[124] Henry Zhu, Abhishek Gupta, Aravind Rajeswaran, Sergey Levine, and Vikash Kumar. Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3651–3657. IEEE, 2019.

[125] Robert L Zimdahl. Weed-crop competition: a review. 2007.

[126] Patrick Zimmer, Michael Halstead, and Chris McCool. Panoptic one-click segmentation: Applied to agricultural data. *IEEE Robotics and Automation Letters*, 8(5):2478–2485, 2023.

[127] Marie L Zingsheim and Thomas F Döring. What weeding robots need to know about ecology. *Agriculture, Ecosystems & Environment*, 364:108861, 2024.

List of Figures

List of Tables