UNIVERSITY OF TRENTO

Department of Information Engineering and Computer Science
Department of Industrial Engineering
Fondazione Bruno Kessler

Doctorate Program in Industrial Innovation - Cycle 38

Innovative Artificial Intelligence / Machine Learning (AI/ML) solutions for smart harvesting of high value crops

Supervisors:
Prof. Farid Melgani
Prof. Ilaria Pertot
Dr. Cesare Furlanello

Author:
Fabio Taddei Dalla Torre

Academic Year 2024/2025

Abstract

This thesis explores the integration of artificial intelligence (AI)-based perception and low-cost robotic manipulation for the autonomous harvesting of underexplored high-value crops. Two case studies, edible flowers and blackberries, were investigated to evaluate a modular pipeline combining detection, segmentation, pose estimation, and robotic manipulation.

For edible flowers, the proposed framework, FloralAI, combined YOLOv5 detection with the Segment Anything Model (SAM) for zero-shot segmentation and semi-automatic annotation. Pose estimation was performed using principal component analysis (PCA), while a novel plucking-point estimation method was introduced based on inferred flower diameter. This strategy reduced reliance on large annotated datasets and enabled generalization across different flower species.

For blackberries, 3D plucking-point estimation was achieved through ellipsoid fitting combined with PCA-based pose estimation, enabling accurate localization and orientation of berries within dense and irregular canopies. This perception pipeline was coupled with a custom soft robotic gripper and a low-cost 6-DOF arm, leading to the development of the first fully evaluated autonomous blackberry-harvesting prototype. Task-based experiments confirmed successful autonomous picking across berries with diverse orientations and positions, establishing replicable benchmarks for future research in agricultural robotics.

The results highlight that while detection models exhibit moderate cross-crop transferability, pose estimation and manipulation strategies require crop-specific adaptations. The findings also emphasize the interdisciplinary nature of agricultural automation: advances in robotics and AI must be complemented by crop research, including plant architecture, growth patterns, and cultivation methods, to enable scalable robotic harvesting.

Overall, this work demonstrates the feasibility of affordable and adaptable robotic harvesting systems. It outlines key future directions, including large-scale dataset generation, multi-modal sensing, real-world deployment, and crop–robot co-design, thereby contributing to the foundation for more sustainable and resilient food pro-

duction systems.

Keywords: Deep neural networks, computer vision, precision farming, edible flowers

Contents

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence. 7, 16, 17, 35, 48
ANN Artificial Neural Networks. 16
API Application Programming Interface. 70
AS Active Stereoscopy. 31
ATM Attention-to-Mask. ix, 43, 51

CLIP Contrastive Language-Image Pre-training. 5
CLS Classify token. 43
CNN Convolutional Neural Networks. 16, 18, 43
CSP Cross-Stage Partial. 41

DBSCAN Density-Based Spatial Clustering of Applications with Noise. 64
DETR Detection Transformer. 19, 43
DH Denavit–Hartenberg. 72
DINO self DIstillation, NO labels. 5
DOF Degree Of Freedom. 72

FAO Food and Agricultural Organisation. 1

GPT Generative Pre-Trained Transformers. 43
GUI Graphical User Interface. 35

HVC High-value crop. 3, 4, 6, 15

ICP Iterative Closest Point. 20
IEEE Institute of Electrical and Electronics Engineers. vii, 2, 3
IoT Internet of Things. 15, 16
IoU Intersection over Union. 46

LiDAR Light Detection and Ranging. 5, 22, 53
LVT Lite Vision Transformer. 59

mAP mean Average Precision. 46, 48
ML Machine Learning. 16, 17

NLP Natural Language Processing. 42
NMS Non-maximum suppression. 40, 42
NR Newton–Raphson. 70

OWL-ViT Vision Transformer for Open-World Localization. 28, 29

PCA Principal Component Analysis. ix, x, 20, 21, 54–58, 60, 64, 65, 85
PoE Product of Exponentials. 70, 72, 73
PUP Progressive Upsampling. 43

R-CNN Recurrent Convolutional Neural Networks. 18, 21
RF Random Forest. 16
RNN Recurrent Neural Networks. 16
ROS Robot Operating System. 69, 72

SAM Segmentation Anything Model. ix, 5, 19, 30, 43, 44, 47, 52, 85
SETR SEgmentation TRansformer. 43
SPPF Spatial Pyramid Pooling—Fast. 41

TB Task-based Benchmarking. 77–79
ToF Time of Flight. 5

URDF Unified Robot Description Format. 69, 72

ViT Vision Transformer. ix, 18, 43, 44, 51, 52

WoS Web of Science. vii, 2, 3

YOLO You Only Look Once. ix, 18, 19, 21, 39–42, 44, 45, 48, 52, 59, 85

Chapter 1

Introduction

1.1 From Traditional to Modern Agriculture

The Food and Agricultural Organisation (FAO) of the United Nations has projected that global food production will need to increase by approximately 70% by 2050 in order to meet the nutritional demands of a population expected to reach nine billion people [1]. Historically, agriculture has managed to keep pace with the growing demand for food, despite a continuous decline in the share of the population employed in the sector. In Italy, for instance, it is estimated that around 60% of the population was employed in agriculture in 1861. However, following World War II, this figure dropped sharply, falling below 10% by the early 2000s [2]. During the same period, the per capita daily availability of calories increased substantially. At the time of Italian unification, the average daily calorie availability, around 2,500 kcal per person, including alcohol consumption, was only slightly above the threshold for undernutrition (2,000–2,300 kcal/day). By the early 2000s, this figure had risen by approximately 70%, reaching about 3,500 kcal per person per day [3].

This remarkable capacity to meet growing food demands despite a shrinking agricultural workforce has been made possible by continuous technological innovation. As illustrated in Fig. 1.1, agriculture has evolved through successive technological revolutions—from traditional manual labor (Agriculture 1.0), to mechanization (Agriculture 2.0), automation and computerization (Agriculture 3.0), and the integration of digital technologies such as the Internet of Things (IoT), sensors, and cloud computing (Agriculture 4.0). Each phase has progressively enhanced productivity, efficiency, and sustainability while reducing dependence on human labor.

Today, agriculture stands at the threshold of a new transformation, Agriculture 5.0, driven by Artificial Intelligence (AI), robotics, and human–machine collab-

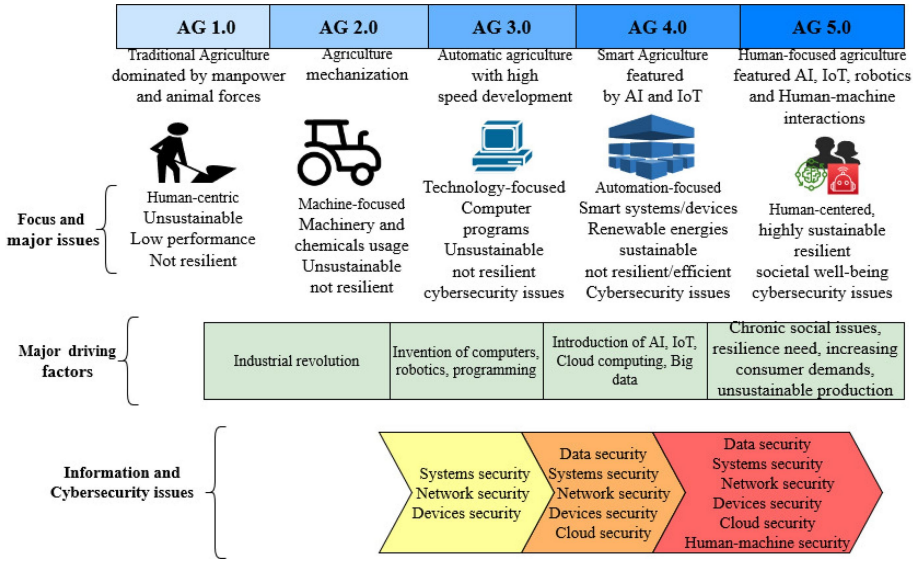


Figure 1.1: Evolution of Agriculture from traditional agriculture to Agriculture 5.0. Figure from Bissadu et al. [4]

oration [4]. These emerging technologies aim to further optimize resource utilization, improve decision-making, and support environmentally sustainable food production, thereby addressing the complex challenge of feeding a growing global population.

Within this context, autonomous agricultural systems, and in particular robotic harvesting technologies, are expected to become central topics of research and development in the coming years. Smart agriculture represents a pivotal approach to tackling two of the sector's most pressing challenges: persistent labor shortages and the growing demand for sustainable farming practices [5].

This trend is mirrored by the increasing attention from both academia and industry. As noted by Hamrani et al. [6], publications on AI and robotics in agriculture were relatively limited before 2018, but a marked surge has been observed since 2019. This growth is illustrated in Fig. 1.2, where red bars represent publications indexed in Scopus, orange bars in the Web of Science (WoS), and blue bars in the IEEE Xplore database.

A similar upward trend can be observed in patent activity. In the field of autonomous devices for precision agriculture, the number of patents increased at an average annual rate of 10.4% between 2017 and 2021 [7]. Together, these academic and industrial developments underscore the growing strategic importance of autonomous agricultural systems and highlight their pivotal role in shaping the future of global food production.

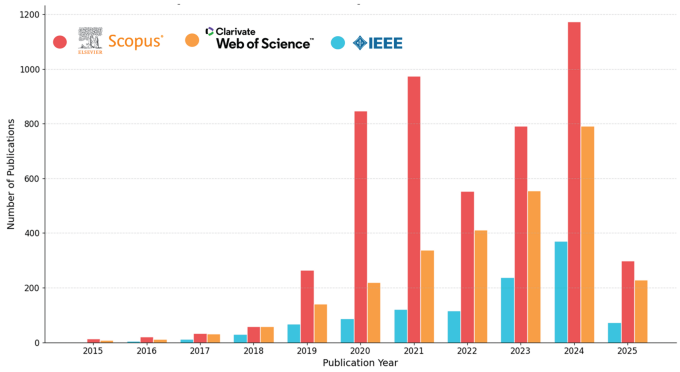


Figure 1.2: Annual publication distribution from 2015 to 2025 across Scopus, WoS, and IEEE. Figure adapted from Hamrani et al. [6].

1.2 Background and Motivation

As agriculture advances into the era of Artificial Intelligence and human–machine collaboration, a critical challenge arises: ensuring that technological innovation is equitably integrated across all segments of crop production. While staple crops such as wheat and maize have undergone profound transformations through mechanization, automation, and data-driven management, high-value crops remain largely dependent on manual labor for key operations. High-value crop (HVC)s are non-traditional agricultural products that offer significantly higher market value per unit area than staple crops. Examples include vegetables, leafy greens, herbs, spices, apples, berries, and edible flowers [8]. A key distinguishing factor between HVCs and traditional staple crops lies not only in their profit margins but also in their level of mechanization. For crops such as wheat, maize, and potatoes, mechanization has been adopted across the entire production cycle, from soil preparation and seeding to harvesting, leading to significant gains in productivity, consistency, and profitability [9, 10, 11]. Studies have shown that mechanization in staple crops can substantially reduce labor costs while improving yield and economic efficiency, particularly through increased cultivation area and more standardized crop management practices [12, 13].

In contrast, the mechanization of HVCs remains limited. Due to the unique morphological structures of many HVCs, such as irregular canopy forms and delicate or easily bruised harvestable parts, the application of high-throughput mechanical systems poses significant challenges [14]. As a result, operations such as harvesting are still predominantly performed manually. This lack of mechanization contributes to high labor requirements and input costs, particularly during harvest, which is consistently reported as one of the most significant contributors to total production

expenses in HVCs [15, 16].

As such, the introduction of advanced technologies for autonomous and intelligent harvesting offers a promising pathway to substantially reduce these costs. Robotic systems have shown potential in reducing labor dependency and operational inefficiencies while maintaining crop quality during harvest [17, 18]. By automating the harvesting process, production costs can be lowered, which may directly influence market dynamics, most notably by reducing the retail prices of high-value crops. Given the high price sensitivity of consumer demand for fresh, nutrient-rich foods, such reductions could improve accessibility [19, 20]. Simultaneously, growers would benefit from expanded market opportunities and increased revenue, as reduced production costs improve profit margins and make high-value produce more competitive in domestic and international markets [21].

Another key driver behind the adoption of robotics and autonomous systems in high-value crop production, particularly in harvesting, is the persistent and worsening of labor shortage. Growers are facing increasing difficulty in sourcing sufficient, reliable, and skilled seasonal workers [22, 23]. This labor scarcity not only disrupts harvesting schedules but also threatens crop quality and overall farm profitability. In this context, automation offers a compelling solution to bridge the labor gap and ensure timely, efficient harvesting operations.

The panorama of High-value crops is quite vaste. While for certain of them such as apples, tomatoes or green peppers, some solutions have been explored and implemented, for some other less relevant crop in terms of market share, no solutions have been explored. Among these crop, the following work will focus on two very different and unique crop: edible flowers and blackberries. Indeed by being the two crops so different this allows us to understand the robustness and generability capabilities of the tested techniques. Moreover, both of these products introducing autonomous harvester will be of huge impact.

1.3 Research Problem

A fundamental challenge in the development of autonomous harvesting systems for high-value crops lies in the lack of research attention dedicated to underrepresented crops. While several robotic harvesting solutions have been developed for mainstream crops such as apples, strawberries, and tomatoes, these systems are typically highly specialized, designed to suit the specific morphological and environmental characteristics of a narrow set of crops [18, 24, 25]. In contrast, many other HVCs remain largely untested, and it is not yet known to what extent existing systems can

be generalized across species with different canopy structures, fruit presentations, or harvesting conditions.

This issue is especially pertinent with regard to 3D pose estimation, which is a core capability for robotic harvesting. Accurate pose estimation allows a robot to localize and orient the target produce in complex environments. While recent studies have achieved promising results in crops such as grapes, strawberries, and tomatoes using deep learning and point-cloud fusion [26, 27, 28], the robustness and adaptability of these models across diverse crops has not been fully explored. Only a few recent efforts, such as those focusing on blackberries and roses, have attempted to extend these techniques to underrepresented crops [29, 30].

Another critical challenge in the development of autonomous harvesting systems lies in their perception. These systems rely heavily on computer vision architectures and sensor fusion strategies, which typically combine RGB imagery with depth data from stereo or Time of Flight (ToF) cameras, and in some cases, Light Detection and Ranging (LiDAR) or hyperspectral sensors, to generate a 3D understanding of the scene [31, 28, 32].

Despite recent advances in deep learning, one of the most significant limitations remains the dependence on large volumes of high-quality annotated training data. Producing such data manually is both time-consuming and costly. While synthetic data generation has been proposed as an alternative, its effectiveness in generalizing across diverse crop types and real-world conditions is still limited and requires further validation [30].

To address this, recent research has turned to semi-automatic annotation pipelines powered by foundation models such as the SAM, self DIstillation, NO labels (DINO)v2, and Contrastive Language-Image Pre-training (CLIP). These models have demonstrated strong zero-shot and few-shot capabilities, making them well-suited to assist in dataset creation with minimal manual supervision [33].

Addressing these gaps, particularly by evaluating pose estimation techniques on morphologically diverse crops and developing new semi-automatic annotation pipelines using foundation model architectures, is essential for broadening the impact, scalability, and practical applicability of autonomous harvesting technologies in high-value crops.

1.4 Research Challenges and Open Issues

Despite recent advancements in autonomous harvesting systems, several critical challenges remain open. One major issue is the high degree of variability in field conditions, including factors such as occlusions from foliage, inconsistent lighting, and complex backgrounds. These elements significantly reduce the robustness and reliability of computer vision models that perform well under controlled conditions. Another key challenge is the seasonality of crop availability, which limits both data collection and testing opportunities to narrow time windows during the growing season. This not only slows down development cycles but also complicates the transition from laboratory testing to real-world deployment, as domain adaptation between in-lab and in-field scenarios remains non-trivial.

Additionally, there is an ongoing need to design and engineer crop-specific end-effectors. As demonstrated in this work through the implementation of a soft inflatable gripper tailored for blackberry harvesting, a one-size-fits-all approach is currently not feasible. Different crops present varying physical characteristics—such as size, shape, fragility, and detachment resistance—which necessitate customized mechanical solutions to ensure safe and effective harvesting. Addressing these challenges is essential for advancing the scalability, generalizability, and practical deployment of autonomous systems in high-value crop production.

1.5 Research Objectives

The primary objective of this work is to design, develop, and validate a complete robotic harvesting pipeline capable of autonomously executing the key stages of crop perception and manipulation. In addition to achieving reliable harvesting performance, particular emphasis is placed on ensuring the pipeline's adaptability and transferability across morphologically diverse high-value crops. To this end, the research focuses on evaluating the generalizability of 3D pose estimation techniques across two representative case studies, blackberries and edible flowers. In parallel, the study investigates the integration of foundation models into semi-automatic annotation pipelines to accelerate dataset generation and minimize manual labeling efforts. Collectively, these research directions aim to advance the development of scalable, versatile, and efficient robotic harvesting systems that can be applied to a broad spectrum of underrepresented HVCs.

1.6 Contributions of this Work

This research presents the development of an Artificial Intelligence (AI)-based vision framework designed to support robotic harvesting across a variety of high-value crop species.

A major contribution of this work is the integration of multiple AI modules, specifically for object detection, 3D pose estimation, and plucking point estimation, into a unified and modular pipeline capable of operating effectively across morphologically diverse crops. The framework leverages machine learning methods that require minimal or no additional training data, relying on few-shot and zero-shot approaches to enable rapid deployment across new crop types. These approaches are also applied to the development of a semi-automatic annotation pipeline, which harnesses the generalization capabilities of foundation models to generate high-quality labeled datasets, significantly alleviating the burden of large-scale manual annotation. Additionally, a novel plucking point estimation method is introduced, based on indirect inference from a curated dataset of flower morphology measurements.

A further key contribution of this work is the design, implementation, and evaluation of a complete robotic harvesting system tailored for blackberry picking. This system incorporates a custom soft inflatable gripper, a low-cost 6-degree-of-freedom robotic arm, and a dual-camera vision-based control strategy. The full pipeline has been systematically tested under realistic conditions, and its performance has been evaluated using objective, structured metrics. The results demonstrate not only the feasibility of deploying affordable and adaptable robotic harvesting solutions for underrepresented high-value crops but also provide a benchmark methodology for future research. This benchmark aims to support reproducibility and foster the development of standardized evaluation protocols within the domain of agricultural robotics.

Chapter 2

Modular Harvesting Framework

2.1 System Architecture Overview

The harvesting framework was implemented using a modular architecture to ensure adaptability and support a structured development process. The work focuses on two key crops: edible flowers and blackberries. These were chosen not only because they are high-value crops (HVCs), but also because their long harvesting periods provide ample time for data acquisition and system testing without being constrained to a narrow time window.

In addition, the two crops are morphologically quite different, which allows for testing the generalizability of the system. Edible flowers, in particular, encompass a wide variety of species, resulting in significant variability and further emphasizing the need for a flexible and easily adaptable framework. This modular approach ensures both adaptability across diverse crop morphologies and a structured workflow that facilitates linear development and effective debugging.

As illustrated in Fig. 2.1, the pipeline begins with data acquisition, achieved either through targeted collection campaigns or by integrating publicly available datasets, followed by refinement of object detection architectures and, where required, segmentation. The subsequent module performs 3D pose estimation by combining RGB imagery with point cloud data to accurately localize and orient target crops. Building on this perception layer, the final module manages robotic manipulation, executing the harvesting action through crop-specific end-effectors.

This structured design supports the evaluation of generalizability across different crop types and forms the foundation for the robotic prototypes developed and experimentally validated in this study. The following sections describe in detail the main modules of the framework, focusing on the vision and manipulation components that

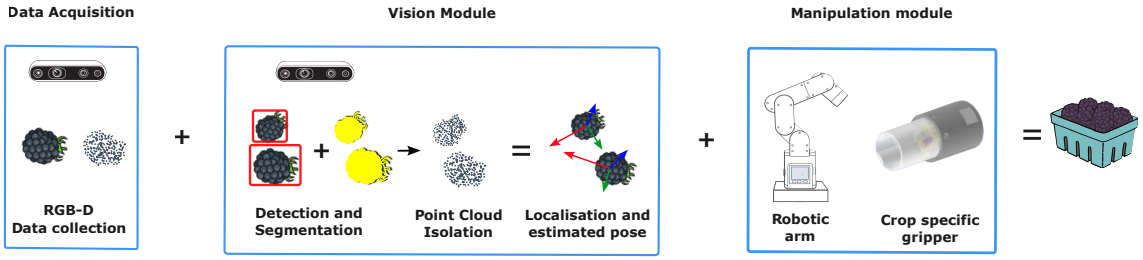


Figure 2.1: Modular harvesting framework with three stages: data acquisition (RGB-D collection and point cloud isolation), vision (detection, segmentation, and 3D pose estimation), and manipulation (robotic arm with crop-specific gripper). Blackberries are shown only as a representative example.

enable perception-driven autonomous harvesting.

2.2 Key Contribution of the Framework

This work advances the state-of-the-art in autonomous harvesting by introducing both methodological innovations and technical implementations across detection, pose estimation, and robotic manipulation.

At the methodological level, the framework integrates multi-modal perception by combining RGB and depth information in a unified pipeline that performs detection, segmentation, and 3D pose estimation. To improve robustness and adaptability, task-specific detection models, namely FLOLO and Berr-YOLO, were developed using multi-stage transfer learning and domain-adaptive fine-tuning. Additionally, the framework leverages foundation models such as the Segment Anything Model (SAM) to enable zero-shot segmentation and streamline annotation processes. To ensure consistent instance-level segmentation in cluttered scenes, a mask-selection strategy based on centroid proximity was implemented. Complementing these perception innovations, a modular and sequential approach for approach and detachment motion planning was designed to guarantee safe, collision-free harvesting.

From a technical perspective, the framework features a perception pipeline capable of estimating six-degree-of-freedom (6-DoF) object poses for a wide range of crops. Iterative inverse kinematics strategies were developed for both approach and detachment trajectories, including mechanisms to handle kinematically unreachable poses. The system demonstrates real-time performance using RGB-D sensing in combination with YOLO-based detection, maintaining robustness even when exposed

to out-of-domain images. Moreover, the modular architecture allows the seamless incorporation of new crops, end-effectors, or sensing modalities without the need for significant reengineering. Collectively, these contributions establish a flexible, scalable, and practically feasible framework for perception-driven autonomous harvesting.

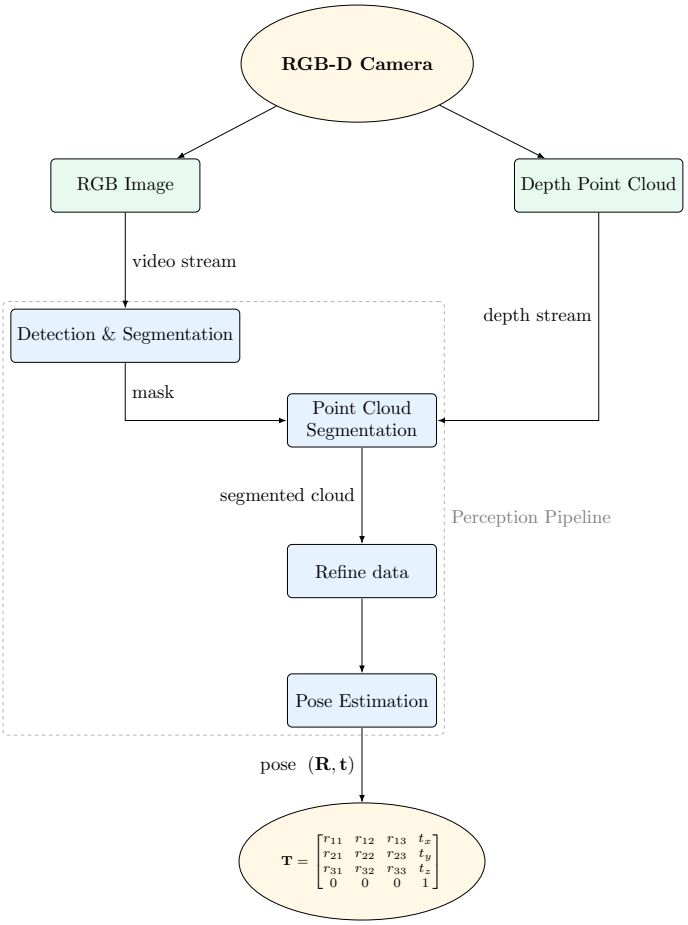


Figure 2.2: Overview of the perception pipeline for target pose estimation. The RGB-D camera provides synchronized color and depth data, which are processed through detection, segmentation, and point-cloud refinement stages to estimate the object pose represented by the homogeneous transformation matrix $\mathbf{T}$.

2.3 Vision Module

The vision module forms the perception layer of the harvesting framework. Its purpose is to detect and localize target crops by estimating their six-degree-of-freedom (6-DoF) pose in the robot's coordinate frame. As shown in Fig. 2.2, the perception pipeline begins with RGB-D data acquisition. The RGB stream is processed by a

detection and segmentation network that isolates the target crop within the image plane. The segmentation mask is then used to filter the depth map, generating a corresponding 3D point cloud of the target item.

Subsequent point-cloud refinement removes noise and outliers, ensuring an accurate geometric representation of the target. Finally, the pose estimation module computes the object's transformation with respect to the camera frame, represented by the homogeneous transformation matrix $\mathbf{T}$:

$$\mathbf{T} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The estimated transformation provides the position and orientation of the target, which serve as the input for the manipulation module.

2.4 Manipulation Module

The manipulation module governs the harvesting action by transforming the estimated target pose $\mathbf{T}$ into executable motion commands for the robotic arm and its end-effector. As illustrated in Fig. 2.3, this module generates both the approach and detachment trajectories required for safe and efficient item retrieval.

The key transformation matrices are resolved through an inverse kinematics (IK) algorithm, which computes the corresponding joint-space coordinates of the manipulator. Specifically, $\mathbf{T}$ represents the estimated rotation and translation $(\mathbf{R}, \mathbf{t})$ of the approach point toward the target item. To model the detachment motion, a sequence of translated poses $\mathbf{T}'$ is generated along the local picking axis, defined as:

$$\mathbf{T}' = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x + l\, r_{11} \\ r_{21} & r_{22} & r_{23} & t_y + l\, r_{21} \\ r_{31} & r_{32} & r_{33} & t_z + l\, r_{31} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where l denotes the translation distance along the local end-effector axis, approximately corresponding to the characteristic length of the item being harvested. The resulting configurations derived from $\mathbf{T}$ and $\mathbf{T}'$ are used as inputs for the path-planning and detachment-planning submodules.

Both submodules operate iteratively. The approach path submodule refines the

trajectory across a set of interpolated points within the planned path, ensuring smooth and collision-free motion. The detachment submodule performs iterative planning over the points defined in the detachment trajectory, which are determined from the inferred spatial coordinates of the target. The detachment phase ("Grab Item") is executed only after the approach phase ("Approach Item") has been successfully completed, enforcing a safe and sequential manipulation process.

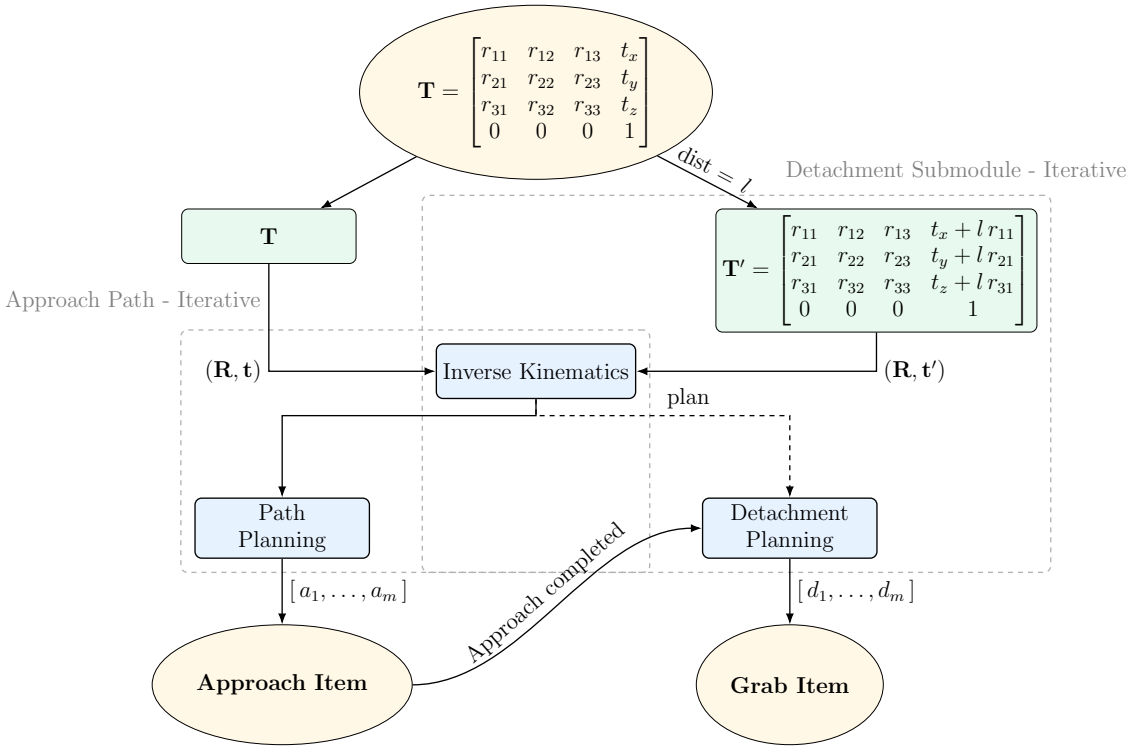


Figure 2.3: Overview of the manipulation architecture. The system computes the approach and detachment trajectories from the approach pose $\mathbf{T}$ and a sequence of translated poses $\mathbf{T}'$ generated along the picking axis. The approach and detachment submodules operate iteratively: the approach path is refined over the number of interpolated points in the planned trajectory, while the detachment process iterates over the points defined in the item's detachment planning. The resulting coordinate sequences are used to execute the approach and grasp actions.

It is important to note that, during computation of the approach path, the inverse kinematics solver may occasionally fail to find a collision-free solution. Such failures can occur when the pose $\mathbf{T}$ is kinematically unreachable due to self-collision, interference with external objects, or when it lies outside the manipulator's workspace. In these cases, the algorithm automatically skips the current instance and proceeds to process the subsequent one. A similar situation may arise during computation

of the detachment path; however, since inverse kinematics is applied to configurations that are spatially close, typically separated by only a few millimeters, a valid solution is almost always found.

Chapter 3

Background and State of The Art

3.1 Smart Agriculture and Automation

The agri-food sector is currently confronted with a set of unprecedented challenges. On one hand, global food production must increase to meet the needs of an expanding population. On the other hand, consumer preferences are shifting toward healthier and more sustainable dietary choices [34]. Concurrently, the sector is constrained by a persistent shortage of skilled and reliable labor [35].

Technological innovation has the potential to address, or at least mitigate, the pressing challenges in agriculture related to productivity, sustainability, and labor shortages [36, 37]. Indeed, since the introduction of synthetic molecules, agriculture, and in paticular HVCs have not experienced a significant technological revolution [38, 39, 34]. One of the main obstacles lies in the complexity of agricultural environments: fields are inherently unstructured, with uneven terrain, variable crop and canopy architectures, and constant exposure to unpredictable weather. These conditions complicate machine operation and reduce the reliability of sensor-based systems [40, 18]. Recent advances, however, are beginning to overcome these challenges by making technologies more robust and field-ready. For instance, Internet of Things (IoT)devices can now collect and process in-field data in real time, enabling farmers to make data-driven decisions that improve efficiency and reduce resource waste [41]. Similarly, remote sensing technologies, particularly satellite-based monitoring of parameters like soil moisture, canopy exposure, and microclimate conditions, are increasingly integrated into precision farming workflows to optimize irrigation, fertilization, and crop management [42].

In addition to sensing and decision-support technologies, robotics is becoming central to precision agriculture. Collaborative robots, for instance, are being de-

ployed in harvesting systems such as strawberry or raspberry picking, where robotic carts transport yields and relieve human workers from physically demanding tasks [43, 44]. This not only improves labor efficiency but also reduces the time harvested fruit remains in the field, thereby enhancing postharvest quality [35]. Autonomous harvesting robots, although promising, remain a technical challenge due to the delicacy of fruit handling and variability in crop conditions [18]. Nevertheless, other robotic applications, such as autonomous sprayers, are increasingly adopted. By dynamically adjusting spraying patterns in response to canopy density, wind conditions, and vehicle speed, these systems reduce the use of phytosanitary products while improving the timeliness and effectiveness of treatments [34, 45].

Taken together, these advancements illustrate a paradigm shift in agriculture: from mechanization as a means of labor substitution toward AI-driven automation and robotic systems that enhance precision, sustainability, and resilience in food production [41]. These emerging technologies increasingly rely on AI and ML techniques, which are now central to agricultural innovation

3.2 Artificial Intelligence and Machine Learning in Agriculture

In addition to sensors and automation, recent advances also focus on the use of AI and Machine Learning (ML) algorithms. These systems can be integrated with the previously mentioned IoT devices, satellites, and other sensing technologies to provide not only data insights but also predictive analytics [46, 47].

For instance, ML models have been widely applied to crop yield prediction, combining historical datasets with sensor and satellite data. Muruganantham et al. [48] provide a systematic review of ML and deep learning methods for yield forecasting, while Joshi et al. [49] emphasize that models such as Random Forest (RF), Artificial Neural Networks (ANN), and Convolutional Neural Networks (CNN) are particularly effective for maize and soybean. Deep learning approaches, such as the CNN–Recurrent Neural Networks (RNN) framework proposed by Khaki and Wang [50], further demonstrate how yield can be forecasted with improved accuracy and robustness. Such predictions are essential for farmers in managing risks associated with pests and diseases, as well as in planning optimal harvest strategies. Yield forecasts support decisions on whether fruit thinning is needed, or when to harvest, in order to maximize profits according to market absorption capacity. This predictive capability also contributes to better logistics management and demand forecasting, thereby reducing post-harvest losses.

Timely treatment of pests and diseases is another critical factor in sustainable farming, as it minimizes the use of phytosanitary products while increasing treatment effectiveness. AI and ML systems can be deployed for both the detection and forecasting of pest and disease outbreaks [51, 52, 53]. For example, in the case of apple scab, predictive models use weather data and crop phenology to estimate both the timing and severity of infections [54, 55]. This enables farmers to apply treatments at precisely the right moment. In addition, computer vision algorithms can be used to identify pests directly from crop imagery, providing another layer of protection [56].

Further applications include weed detection, where computer vision techniques allow for accurate identification of weed species, supporting efficient and targeted weed management strategies [57, 58]. Similarly, AI-powered precision spraying systems can map crop canopy structures or identify specific treatment zones, ensuring that phytosanitary products are applied exactly where needed [57]. A key enabler of these AI and ML applications is computer vision, particularly through object detection and segmentation

3.3 Vision-Based Detection and Segmentation

Vision-based algorithms constitute a cornerstone in the advancement of autonomous agricultural systems. By enabling robots and AI/ML models to perceive and interpret complex field environments, they provide critical inputs for downstream tasks such as navigation, fruit harvesting, weed and pest detection [59, 60]. Since the early stages of AI and ML research, computer vision has been a central focus, particularly in the areas of object detection and image segmentation. Object detection involves identifying and localizing specific targets within images or video streams [61], while segmentation goes a step further by delineating the precise boundaries and shapes of those targets within a scene [62]. Together, these techniques form the basis for high-resolution perception, which is indispensable for reliable and efficient automation in agriculture [63].

3.3.1 Object Detection

Identifying and localizing objects in images or video streams is a fundamental task in computer vision. Unlike simple classification, which only determines what is present in an image, object detection provides the additional and crucial information of where the object is located. This spatial understanding is pivotal for agricultural

tasks such as weed detection, fruit picking, and yield estimation [42].

Early approaches relied on standard classification combined with sliding window techniques, typically implemented in a two-stage process. In this framework, a window is systematically moved across the image in a grid-search manner, and for each window, a classification algorithm is applied. By merging the classification results with the corresponding window coordinates, the system could determine both the identity and location of objects. These methods were soon enhanced by the introduction of CNN, which are particularly suited for image processing [64]. CNNs employ convolutional kernels to automatically extract hierarchical features from images, while also exhibiting translation, rotation, and scale invariance. These properties are crucial in agriculture, where the same object, such as a fruit or a leaf, may appear in varying positions, orientations, or sizes. The combination of sliding window techniques and CNNs led to the development of region-based methods such as Recurrent Convolutional Neural Networks (R-CNN) and its improved variants, including Fast R-CNN [65, 66].

Despite these advancements, sliding window-based approaches presented significant drawbacks. Scanning the entire image with multiple windows is computationally expensive, making real-time applications impractical. Moreover, determining the appropriate window size is problematic: windows that are too small could miss objects, while excessively large windows could include irrelevant background information. Overlapping windows also often resulted in redundant detections. To overcome these limitations, the You Only Look Once (YOLO) family of algorithms introduced a paradigm shift in object detection. Instead of scanning multiple windows, YOLO treats detection as a single regression problem, directly predicting bounding boxes and class probabilities from the entire image in one forward pass of the network. This design dramatically improves speed, enabling real-time performance without sacrificing much accuracy [67, 68, 69]. YOLO's architecture divides the image into a grid, and each grid cell is responsible for predicting bounding boxes and class probabilities. This allows the network to simultaneously detect multiple objects across the scene while maintaining high efficiency. For agricultural applications, YOLO has proven especially valuable in tasks such as fruit counting, crop monitoring, and weed identification, where real-time feedback is essential for autonomous field robots[70, 71]. However, a notable drawback of YOLO is its difficulty in accurately detecting small objects, since fine details may be lost when an image is divided into grid cells [69].

More recently, ViT have emerged as a powerful alternative to CNN-based models. Inspired by transformers in natural language processing, their core component is the self-attention mechanism, which captures both local and global contextual re-

lationships without relying on convolutional layers [72, 73]. A notable architecture is Detection Transformer (DETR), which directly predicts a fixed set of bounding boxes and class labels in an end-to-end pipeline [74]. While DETR streamlines the detection process and performs well on complex scenes, it remains data-hungry and computationally demanding. In practice, large annotated datasets are required for effective training, which represents a significant challenge in agriculture, where labeled data is often scarce.

3.3.2 Object Segmentation

Segmentation extends the concept of object detection by moving from bounding-box localization to pixel-wise classification of an image. In segmentation, each pixel is assigned to a specific class, enabling not only the identification of objects but also the precise delineation of their shapes and boundaries. This level of detail is crucial in applications where exact geometry matters, such as medical diagnosis [75], autonomous driving [76], or robotic manipulation [77], where knowing how an object is shaped and how it interacts with its environment directly impacts decision-making and task execution.

Achieving such precision requires specialized model architectures. One of the most influential is the U-Net, originally proposed by Ronneberger et al. [75]. Its characteristic U-shaped structure consists of a contracting path, which captures hierarchical image features, and an expanding path, which reconstructs a segmentation mask while incorporating spatial details through skip connections. This design enables highly accurate localization even with limited training data, making the U-Net a cornerstone for segmentation tasks.

More recently, segmentation capabilities have also been incorporated into object detection frameworks. From YOLOv8 onwards, the architecture has been extended to perform instance segmentation, combining real-time detection with pixel-level precision. This allows for fast and efficient segmentation for real-time applications [78].

Another breakthrough is the SAM, introduced by Meta AI in 2023 [33]. SAM is based on a ViT backbone, combined with a prompt encoder and a lightweight mask decoder. SAM is a foundation model trained on a massive dataset of diverse images and masks, enabling it to generalize to a wide range of segmentation tasks with minimal or no fine-tuning. This dramatically reduces the reliance on large, annotated datasets while still delivering high-quality results. Such flexibility accelerates the deployment of segmentation models in the field, where adaptability and efficiency

are crucial.

3.4 3D Pose Estimation Techniques

Pose estimation is one of the most critical and delicate components in robotic harvesting, as the success of precise and safe fruit picking depends heavily on accurate localization and orientation determination of the target object [79]. A wide range of techniques have been developed to tackle this challenge. Traditional methods include 3D model matching and registration [79, 80], as well as statistical approaches such as Principal Component Analysis (PCA), which extracts dominant spatial directions of point distributions to approximate object orientation [81]. More recently, the field has expanded to incorporate keypoint-based geometric methods and data-driven deep learning approaches, offering robust solutions under complex field conditions.

3.4.1 Model-based Geometric Methods

Geometric methods estimate pose by directly analyzing the 3D structure of fruits, usually obtained from point clouds or depth images. These approaches rely on aligning observed shapes with reference models or extracting geometric features that capture the object's orientation.

For example, Eizentals and Oka [80] applied Iterative Closest Point (ICP) matching, in which the scanned fruit surface is iteratively aligned with a stored 3D model until the shapes overlap, yielding both position and orientation. Similarly, Guo et al. [79] estimated fruit orientation by matching the observed point cloud with an offline 3D reconstructed model. In addition to precise pose estimation, this approach supports adaptable grasp planning, as the offline stored model reveals feasible contact regions and guides the gripper's alignment to different fruit shapes and sizes.

3.4.2 Shape and Axis-Based Methods

Other methods exploit structural regularities rather than full model matching. Li et al. [82] estimated the symmetry axis of sweet peppers, assuming that the fruit's long axis reveals its orientation, a strategy that avoids the need for detailed modeling. PCA-based approaches take this idea further: by analyzing how point clouds are distributed, they extract the dominant direction of variation, which typically corresponds to the fruit's main growth axis, as demonstrated in tea harvesting systems [81]. Hussain et al. [83] demonstrated a similar idea for apple thinning by detecting

and segmenting small fruits via Mask R-CNN and pairing it with PCA for pose estimation, achieving orientation accuracy within 30° of the ground truth in most cases.

3.4.3 Keypoint-Based and Learning-Driven Methods

Keypoint-based techniques exploit the idea of detecting biologically meaningful landmarks, such as fruit tips, stems, or sepals, from which the fruit's pose can be inferred. For instance, Zhang et al. [84] developed a network to identify keypoints on tomato bunches, and then used their spatial configuration to determine the arrangement and orientation of individual fruits within a cluster. Similarly, Jang and Hwang [85] exploited the relationship between the tomato body and its sepals, treating these natural reference points as cues for estimating fruit poses even under partial occlusion.

Zhao et al. [86] introduced an enhanced version of YOLOv7, named YOLOv7-hv, specifically developed for estimating the 6D pose (position and orientation) of cucumbers. Their approach integrates multiple outputs—fruit detection, instance segmentation, and keypoint prediction—into a unified framework. The 6D pose is obtained by projecting the detected 2D keypoints into the 3D point cloud, which allows precise localization of both the fruit and the cutting point, as well as accurate reconstruction of the complete 3D pose.

Similarly, Tafuro et al. [87] addressed the same challenge in the context of strawberries. Their system, built on Detectron-2 with Mask R-CNN extended for keypoint detection, was able to localize strawberry picking points and estimate fruit orientation accurately. The orientation is inferred from the relative arrangement of detected keypoints, which defines the strawberry's main longitudinal axis. To enhance robustness, PCA is further applied to verify geometric consistency and refine the orientation estimation.

3.5 Robotic and autonomous systems in Agriculture

Robotics and autonomous systems have advanced dramatically over the past decade and are increasingly being applied in agriculture [88]. As highlighted by Duckett et al. [89], autonomous systems are now widely used for a variety of farming operations, particularly in autonomous tractors and precision spraying. In contrast, autonomous

harvesting technologies remain less commonly deployed.

These developments have been largely driven by progress in navigation systems, especially in GPS and vision-based navigation. GPS-guided navigation has become sufficiently precise to be reliably employed in agriculture, with reported positioning errors ranging from 20.1 cm to as little as 2.35 cm [90]. For other applications such as weeding and precision spraying, vision-based methods are more suitable due to their higher accuracy. For instance, Ramya et al. [91] proposed a method in which crop rows are identified using a segmentation technique, while a tracking algorithm maintains row alignment across consecutive frames.

Other approaches exploit different sensing modalities. Jiang et al. [92], for example, employed LiDAR to capture information about the environment. The data were processed using the DBSCAN algorithm to identify tree trunks, K-means clustering to distinguish between left and right rows, and the RANSAC algorithm to fit boundary lines around the trunks. From this, the system computed the main trajectory line for navigation.

3.5.1 Manipulation and Perception

While autonomous navigation has made significant progress, harvesting remains a major challenge that has not yet been efficiently solved. This difficulty arises primarily from the complexity of manipulating harvestable crops, which is affected by factors such as fruit occlusion, unstructured environments, and varying levels of fruit stiffness [93]. In general, applications that involve large-scale interactions with crops, such as precision spraying, have advanced considerably. By contrast, tasks requiring careful and delicate interaction with individual plants, such as harvesting or fruit thinning, have seen far less progress. The core challenge lies in robotic manipulation: actions must be performed both rapidly, to ensure efficiency, and delicately, to avoid damaging the plants or fruits [94].

While kinematics algorithms for robotic manipulation have been extensively studied across domains, perception remains a rapidly evolving area in agricultural robotics. In particular, visual servoing has emerged as a powerful approach for improving manipulation, benefiting from advances in RGB and RGB-D cameras as well as modern vision architectures [95]. These improvements parallel broader developments in agricultural robotics that emphasize robust perception as a foundation for effective control [94].

Within visual servoing, researchers typically distinguish between two sensor configurations: eye-in-hand and eye-to-hand (eye-on-base). In the eye-in-hand config-

uration, the camera is mounted near the end-effector, providing close-range, high-resolution views of the target crop. This arrangement enables precise interaction but sacrifices an overview of the wider scene. Conversely, eye-to-hand configurations mount the camera on a fixed base, producing a stable frame of reference and a larger field of view that may include both the plant and the manipulator itself [15, 17]. Each configuration has been tested in agricultural contexts, with eye-in-hand proving useful for fine manipulation tasks such as grasping or cutting, while eye-to-hand is more effective when global scene awareness is required, for example in obstacle avoidance or path planning [95].

Both setups enable visual servoing control, where manipulator trajectories are generated using visual feedback. Two principal strategies are commonly applied: open-loop and closed-loop servoing. In the open-loop approach, once the target position has been identified, the manipulator moves to that location using kinematic and path-planning algorithms, but without continuous error correction. This strategy has been applied in fruit harvesting prototypes, where system simplicity and speed were prioritized [15]. In contrast, closed-loop servoing continuously updates the end-effector position using real-time vision feedback, allowing the robot to compensate for disturbances such as branch motion or partial occlusions. This approach has shown higher robustness in unstructured environments such as orchards and greenhouses [17, 95]. As highlighted by recent reviews, the integration of closed-loop vision with advanced kinematics is likely to be crucial for achieving reliable manipulation in agricultural settings [94].

3.5.2 End-Effector and Gripper Design

Once a robotic system has identified and localized the target item, it requires an end-effector—the terminal component of the manipulator responsible for actively interacting with the environment and executing the task. In agricultural robotics, this device is commonly referred to as a gripper, since the primary goal in many applications is to grasp and hold fruits during harvesting. A wide variety of designs have been explored. For instance, Elfferich et al. [96] developed a twisting-tube gripper made from cloth material that gently wraps around blackberries, providing a firm yet delicate grip. Lazo et al. [97] proposed a gripper capable of simultaneously cutting and holding the stalk of peppers, while Parsa et al. [98] designed a similar device for strawberries, equipped with a stereo RGB-D camera and two additional fingers specifically for handling occlusions. Fan et al. [99] employed a standard three-finger gripper for apple harvesting.

This diversity of designs highlights that no single universal gripper can meet the

requirements of all crops or tasks. Instead, progress in agricultural robotics strongly depends on the development of crop- and task-specific end-effectors that combine delicacy, efficiency, and adaptability to the unique characteristics of each harvesting scenario.

Chapter 4

Materials

4.1 Edible Flowers Dataset

4.1.1 FloraDet a custom flower detection and segmentation dataset

The dataset was built with images of flowers collected in a commercial greenhouse located in Mira (VE), Italy, and managed by L'Insalata dell'Orto s.r.l. [100], a leading company in the edible flower sector. The company specializes in the cultivation, packaging, and distribution of fresh vegetables, salads, and 17 species/varieties of edible flowers. The plants were cultivated on benches measuring 70 cm in height, 110 cm in width, and approximately 50 m in length, as illustrated in Fig. 4.1. This configuration is one of the most widely adopted configurations in greenhouse floriculture. This layout not only standardizes plant positioning but also provides significant advantages for robotic harvesting, such as a uniform working height and a straight-line arrangement that improves the consistency and quality of the collected data.

(a) Marigold (Texana) (b) Snapdragon (Leo) (c) Pansy (Carneval) (d) Horned Pansy (Sorbet)

Figure 4.2: Sample images of the four edible flowers considered in the experiments.

(a) (b)

Figure 4.1: Example of the L'Insalata dell'Orto greenhouse showing cultivation benches used for edible flower production.

Given this arrangement, a perpendicular, top-down acquisition strategy was adopted. Moreover, due to the structural diversity of the crop canopies, alternative viewpoints offered no meaningful advantage, as key floral structures (e.g., the calyx) are often occluded by petals or foliage. In contrast, the top-down approach ensured consistent framing of the full bench width while minimizing background noise and maximizing the visibility of the flowers.

The collected data focused on four species (Fig. 4.2) that represent a substantial portion of the company's production and exhibit distinct characteristics on both flower and plant morphology, providing a diverse sample for testing the implemented techniques: marigolds (*Tagetes patula* cv Texana), snapdragon (*Antirrhinum majus*, cv Leo), pansy and horned pansy (*Viola* x *wittrockiana* cv Carneval and *Viola cornuta* cv. Sorbet, respectively). The inclusion of different flower types introduces variability into the dataset. For instance, snapdragon flowers typically lean to one side, whereas marigolds and pansies generally maintain an upright orientation.

(a) (b)

Figure 4.3: Comparison of marigold flower conditions: (**a**) July vs. (**b**) November.

Table 4.1: Structure of the FloraDet dataset. The number of collected images and number of flowers from those images in the two data campaigns are listed in the three varieties, subdivided on Month sampling.

	Items	**Pansy**	**Marigold**	**Snapdragon**
July	Images	–	25	13
	Flowers	–	854	762
November	Images	64	21	11
	Flowers	375	401	51

To introduce environmental variability into the dataset, particularly with respect to cultivation conditions (Fig. 4.3), two data collection campaigns were conducted on 22 July and 17 November 2023. Although the flowers were grown in a greenhouse where environmental factors are partially controlled, seasonal variations in flowering induction still occurred, primarily driven by photoperiod. Consequently, flower density per square meter for reference varieties such as marigold and snapdragon varied substantially, ranging from 3 to 50 flowers. Both campaigns included images of marigolds and snapdragon, while pansy data were collected exclusively during the second campaign.

In total, the *FloraDet* dataset comprises 134 high-resolution images at 1280×720 pixels, containing $2,443$ flowers, which were used for model development and validation. Flower instances were manually annotated with bounding boxes using the Roboflow online annotation tool (Des Moines, USA) [101], with all labeling performed by a single trained operator to ensure consistency. The dataset is summarized in Table 4.1. According to commercial quality standards, an edible flower is considered ripe and harvest-ready when approximately 70–80% open. Following this criterion, only ripe flowers were annotated, ensuring that the system was trained specifically to detect commercially harvestable flowers.

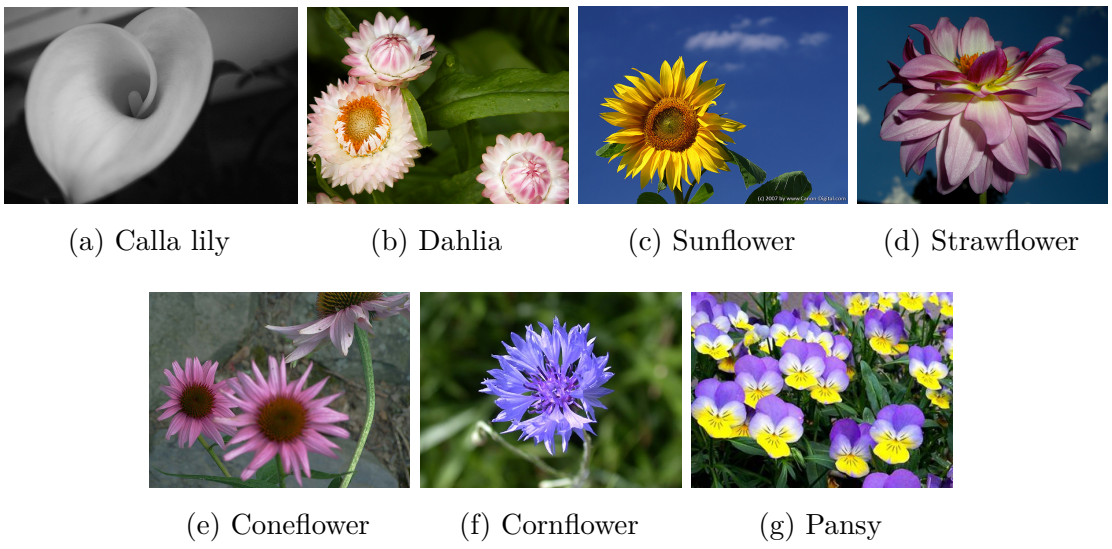
4.1.2 D0 pretraining dataset

An additional *D0* flower dataset was created by merging images from two distinct sources: ImageNet [102] and Kaggle [103]. Notably, the D0 image dataset comprised flower varieties different from those in FloraDet (Fig. 4.4). Specifically, we sourced a set of 1033 images, including flowers from ImageNet, along with their corresponding annotated bounding boxes. This involved retrieving images and annotations from ImageNet Large Scale Visual Recognition Challenge 2010 (ILSVRC2010) [104] of

six different synsets (Table 4.2).

Table 4.2: Common name, ImageNet synset, number of images, and number of flowers in the support D0 dataset from public sources. Pansy images were collected from Kaggle without a specific synset reference.

Common name	**ImageNet synset**	**Number of images**	**Number of flowers**
Sunflower	n11978233	245	294
Calla lily	n11793779	179	233
Cornflower	n11947802	126	146
Dahlia	n11960245	172	191
Strawflower	n11980318	172	242
Coneflower	n11962272	139	159
Pansy	N/A	232	905



(a) Calla lily (b) Dahlia (c) Sunflower (d) Strawflower

(e) Coneflower (f) Cornflower (g) Pansy

Figure 4.4: Examples of the different flowers in the D0 dataset from ImageNet and Kaggle

The D0 dataset was further enriched with images of pansy flowers. A total of 232 images of pansy flowers were obtained from two online datasets: Flower-299 [105] and Flower Color Images [106], both from Kaggle. Since these datasets were originally designed for image classification and lacked bounding boxes, two distinct annotation processes were implemented. The first involved autonomous annotation via zero-shot detection using the open-vocabulary object detection network Vision

Transformer for Open-World Localization (OWL-ViT) [107]. A portion of about 43% ($n = 101$) images were first annotated with OWL-ViT; the remaining $n = 131$ images were manually annotated using the Roboflow annotation tool. This manual annotation process was also beneficial in identifying and removing duplicate images.

4.1.3 Plucking point estimation dataset

To adress the plucking point estimation problem, a dataset of 300 flowers was collected, evenly distributed across marigold, snapdragon, and pansy varieties. Each flower was carefully measured using a caliper, recording the flower diameter and the calyx height.

The measurement distributions for flower diameter (d) and overall flower height (h) are shown in Fig. 4.5. The diameter distribution of pansies exhibits a pronounced right tail, reflecting the considerable size difference between the two species considered (Carneval and Sorbet), with the former being substantially larger. Because the larger pansies had recently been harvested, they are underrepresented in the dataset, further contributing to this skew. In contrast, the distributions for marigolds and snapdragons show less pronounced tails, indicating more consistent flower sizes.

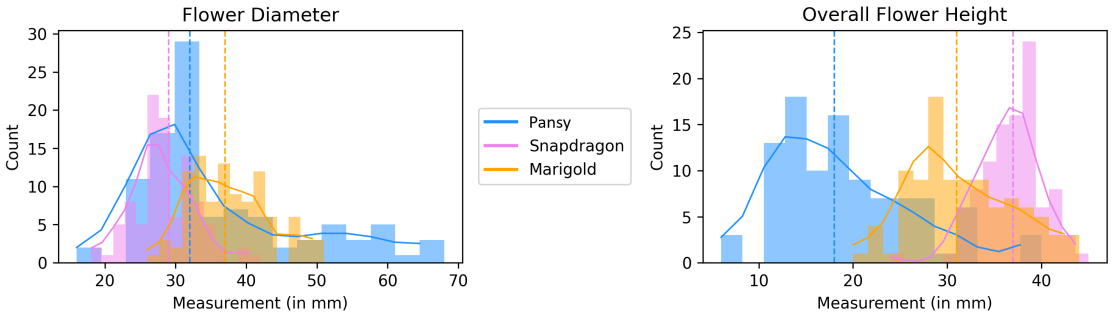


Figure 4.5: Distribution plots for total flower diameter (**left**) and height (**right**). Dotted vertical lines indicate medians, horizontal solid line represents a smooth approximation of the distribution.

4.2 Blackberry Dataset

4.2.1 Pretraining out-of-domain dataset

A dataset for fine-tuning the model on blackberry detection and segmentation was assembled by combining two publicly available datasets (DF1 [108], DF2 [109]) from

Roboflow (Locus Street, USA) [101]. The final dataset comprises 140 images containing 797 annotated blackberry instances.

Since only one of the original datasets included segmentation masks, both datasets were re-annotated using the SAM [33] to ensure consistency and high-quality segmentation across all samples. This produced a unified dataset suitable for both detection and segmentation tasks.

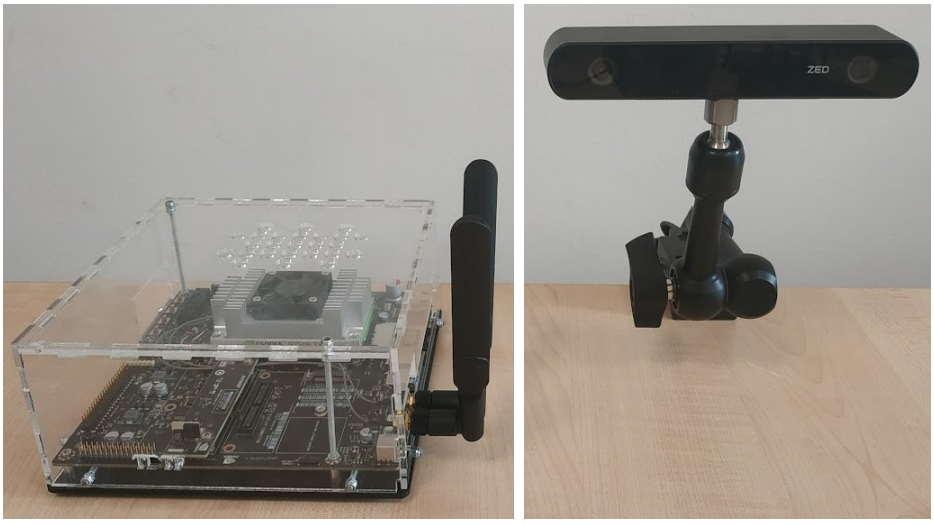
The dataset is highly diverse, representing blackberries under varying conditions and from multiple domains. Figure 4.6 illustrates three examples, highlighting variability in background and fruit presentation.

Figure 4.6: Representative samples from the assembled blackberry dataset, showing variation in background, and fruit presentation.

4.3 Hardware Setup

4.3.1 Sensing Devices

Various sensors were employed across multiple tasks. An Intel (Santa Clara, USA) RealSense D415 RGB–D camera with an approximate field of view of 69° × 49°, two 2,mm lenses, and an image resolution of 1280 × 720 pixels. In addition, two Stereolabs (San Francisco, CA, USA) RGB–D cameras, the ZED2 and ZED2i, were employed. The ZED2 provides a field of view up to 120° using dual 4,MP sensors with 2.1,mm lenses in a native 16:9 format, whereas the ZED2i uses dual 4,mm lenses, an 81° field of view, and integrated polarizing filters that improve image quality under challenging lighting. Both Stereolabs cameras operated at 1280 × 720 resolution. Additionally, a standard USB RGB camera was used specifically the ZXCN WiFi Endoscope Camera, with a resolution of 471x345 pixels.



(a) (b)

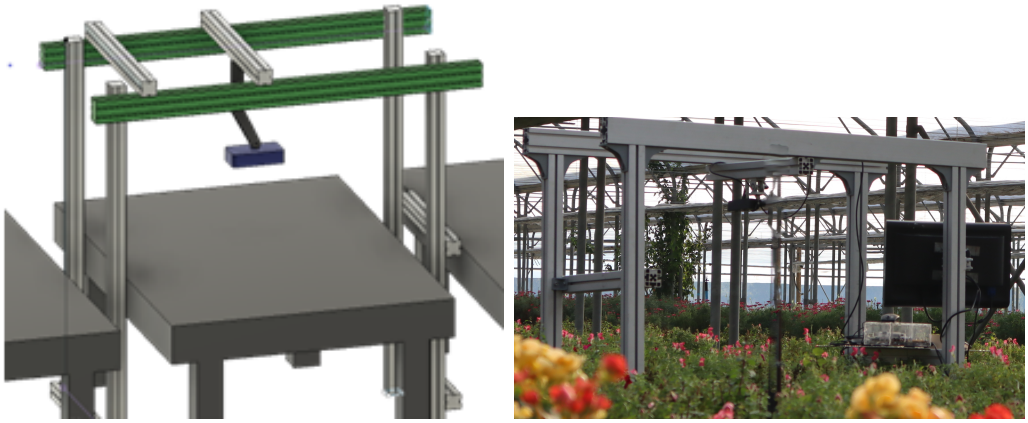
Figure 4.7: Jetson TX2 module (**a**) and Zed2i (**b**) integrated in the data acquisition setup.

The RealSense computes depth via Active Stereoscopy (AS) [110]: it captures left–right stereo images and triangulates matching pixels to estimate depth. An additional unstructured pattern is projected onto the scene to increase texture on low-detail surfaces, refining the correspondence search and thus the depth estimate [111]. Depth is computed directly on the camera's onboard hardware.

By contrast, the Stereolabs cameras are passive stereo systems: depth is obtained solely by triangulating correspondences between the two views. The disparity estimation is performed by the ZED SDK using neural stereo depth methods on a CUDA-capable GPU, which delivers dense depth maps in real time.

4.3.2 Data Collection Cart

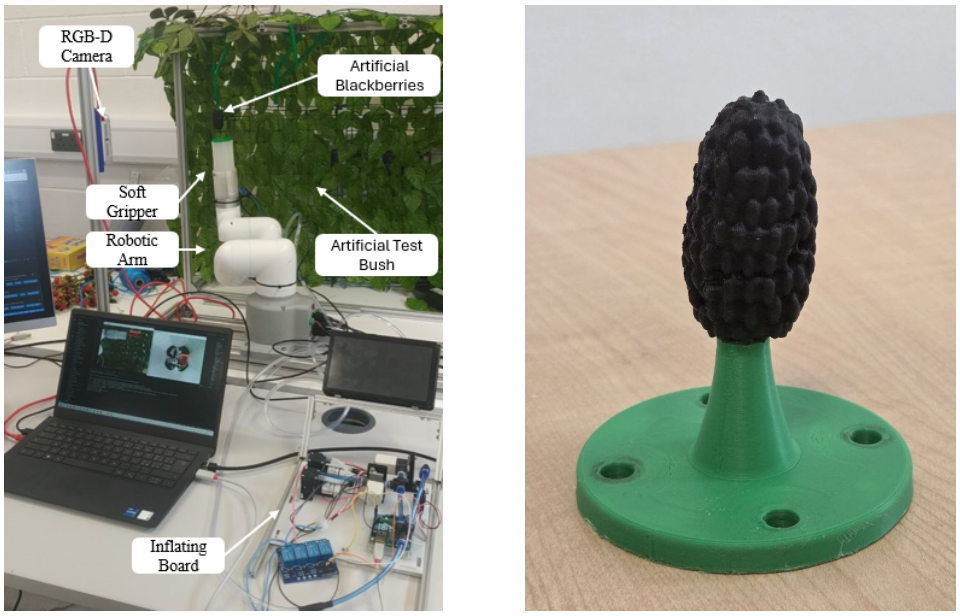
The images for the *FloraDet* dataset were acquired using a custom straddle cart assembled from modular aluminum profiles (Fig. 4.8). The cart measures 120 cm in width, 175 cm in height, and has an 80 cm wheelbase between the front and rear wheels (Fig. 4.8a). It was designed to straddle the benches and be easily pushed or pulled along the rows while carrying the imaging sensors and onboard computing devices. A 3D blueprint of the platform is shown in Fig. 4.8a, and an example of in-greenhouse acquisition is shown in Fig. 4.8b.



(a) (b)

Figure 4.8: Acquisition cart: (**a**) 3D blueprint of the cart; (**b**) example of acquisition in the greenhouse.

4.3.3 Synthetic blackberry harvesting setup



(a) (b)

Figure 4.9: Experimental hardware: (**a**) laboratory setup including a robotic arm, RGB-D camera, synthetic bush, gripper, and computer; (**b**) synthetic blackberry copy of Driscoll's Victoria® variety

To evaluate the framework in pseudo-realistic conditions, a synthetic blackberry bush was constructed. The setup consists of a metal frame with artificial leaves, complemented by plastic adjustable bars that allow positioning of artificial blackberries at arbitrary coordinates (x, y, z) within the reference frame. As shown in Fig. 4.9a, the physical twin is visible in the background alongside the full testing setup. Each berry can also be tilted by an angle $\alpha \in [0, \pi]$ relative to the vertical plane.

The synthetic blackberries (Fig. 4.9b) replicate the size, color, and shape of Driscoll's Victoria® variety, with dimensions of approximately 45 mm in height and 24 mm in width. To mimic the natural detachment resistance of real berries, a small magnet was affixed to the end of each artificial berry at the point where the stem would normally connect. This design allows the robotic arm to detach the fruit from the adjustable bars, ensuring both realistic interaction and experimental repeatability. Although efforts were made to replicate real-world conditions—such as including background elements, a magnetic stem, and adjustable bars for varying orientation and position—the setup still presents some limitations. In particular, it lacks variability in illumination and does not account for occlusions caused by foliage or other fruits.

4.3.4 Soft Gripper

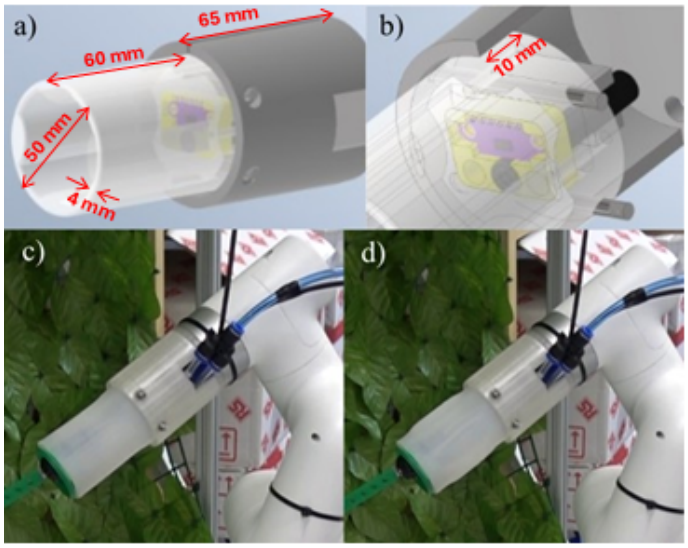


Figure 4.10: CAD views of the gripper (a,b): black cylinder is the USB endoscopic camera. The gripper assembled in deflated (c) and inflated state (d) with an ingested berry.

The gripper used in this work is an enhanced version of the design proposed

by Johnson et al. [112]. It consists of a hollow cylindrical structure with four pneumatically actuated chambers that gently wrap around the berry. When air is pumped into the chambers, the inner walls expand and conform to the fruit, providing sufficient resistance to detach it from the plant while evenly distributing pressure across the surface. This uniform distribution ensures a secure grip without bruising the berry. The gripper is fabricated from Smooth-On Inc. Ecoflex™ 00-50 elastomer.

Compared with the initial prototype described in [112], the new design incorporates an in-palm endoscopic camera (Fig. 4.10a-b), suitable for visual servoing. Moreover, this gripper features thinner silicone walls (4mm for the walls and 2mm for the chambers, down from 10mm), allowing a wider inlet (42mm, up from 24mm) and lower working pressure (from 137kPa to 16kPa). This improved the ingestion of the berry (Fig. 4.10c-d) increased the contact surface, and decreased the power consumption.

4.3.5 Robotic Manipulation Platform

The robotic manipulation system consists of a 6-DOF MyCobot Pro 320 Pi robotic arm (Elephant Robotics, Shenzhen, China). The arm has a radial reach of 350 mm and, according to the manufacturer's datasheet, offers a repeatability of approximately ±0.5 mm. This model was selected as the test platform primarily for its low cost, as high equipment costs are recognized as a major barrier to robotic automation in agriculture [113]. The robotic arm is shown in Fig. 4.9a.

4.3.6 Edge computing and control board

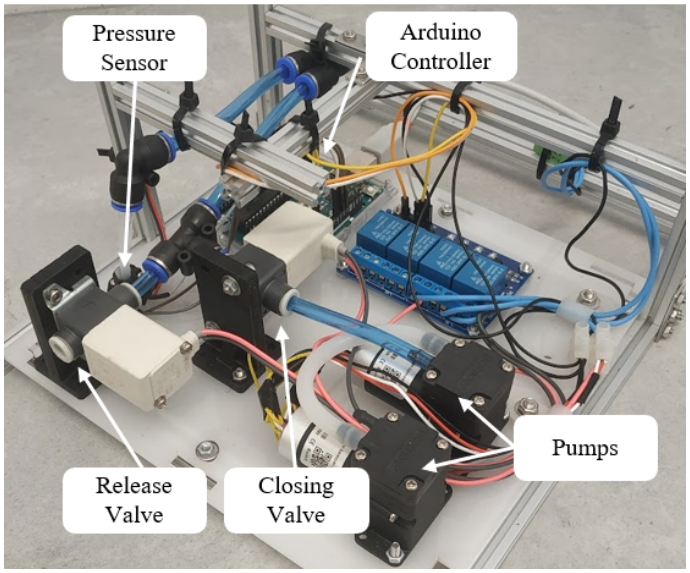


Figure 4.11: Inflation board for controlling gripper actuation. The system consists of an Arduino control unit, inflation pumps, a pressure sensor, and two control valves.

To process data from the Stereolab sensor, an NVIDIA Jetson TX2 module was employed (Fig. 4.7) [114]. Jetson boards are compact system-on-modules designed by NVIDIA (Santa Clara, CA, USA) for executing AI models at the edge, i.e., in real-time operational setups. The Jetson TX2 features a 256-core NVIDIA Pascal™ GPU with CUDA support, a dual-core NVIDIA Denver 2 64-bit CPU, and 8 GB of 128-bit LPDDR4 memory [115]. In this work, the module was used to process images and point clouds of the crop via a Python-based Graphical User Interface (GUI) integrated with Stereolab's Python programming interface.

A custom inflation control board was developed to regulate the soft gripper (Fig. 4.11). The board is based on an Arduino, which, upon receiving an input signal, activates two pumps connected in series while simultaneously monitoring system pressure via a pressure sensor. The inflation process takes approximately 2 seconds, or until the pressure reaches 16 kPa, indicating proper gripper inflation. When a release signal is received, the Arduino opens the release valve, allowing the gripper to deflate. Once deflation is complete, the valve is automatically closed. Because the pumps and valves operate at 12 V, a set of relays was incorporated to amplify the Arduino signal and ensure sufficient power delivery.

Chapter 5

Detection and Segmentation

5.1 Problem Statement

Over the past decades, a wide range of object detection methods and algorithms have been proposed, implemented, and evaluated. Many of the most widely adopted architectures are pretrained on large-scale benchmark datasets such as ImageNet—one of the largest annotated, high-quality image classification datasets available [116], and COCO, a dataset designed for object detection, segmentation, and captioning tasks [117]. Pretraining on such datasets provides these architectures with strong initial feature representations. However, when applied to custom domains, it becomes essential to perform fine-tuning in order to achieve accurate and reliable results. This process mitigates domain shift, a common problem that arises because the pretrained networks are typically trained on data that differ in distribution from the target application.

This adaptation is made possible through transfer learning, a central concept in modern machine learning that enables models to leverage knowledge acquired from one task and apply it to another related task [118]. By exploiting pretrained feature extractors, transfer learning allows architectures to be efficiently specialized to a new domain with considerably less data and computational effort compared to training from scratch. In this sense, fine-tuning not only enhances task-specific performance but also represents a practical and resource-efficient approach to deploying state-of-the-art detection models in specialized applications.

More recently, the emergence of foundation models has introduced an alternative paradigm. Instead of relying exclusively on fine-tuning, these models can often be adapted to a target domain through prompting. A single model trained on a broad spectrum of tasks can deliver high-quality detection or segmentation performance

when guided by prompts, such as point-based inputs or textual descriptions of the object of interest [119]. This approach reduces the need for extensive retraining and broadens the potential applicability of a single model across diverse domains.

Nevertheless, high-quality domain-specific data and annotations remain indispensable. Data annotation is still one of the most resource-intensive steps in computer vision pipelines, and despite advances in automation, it continues to demand significant manual effort. In the present work, a hybrid strategy has been adopted: data were collected and images were manually or semi-automatically annotated for object detection, while foundation models were initially explored as components of the detection pipeline but were ultimately employed primarily as automatic annotation tools.

In addition, careful fine-tuning with out-of-domain datasets was performed, demonstrating the feasibility of this approach in improving performance under distributional shifts. However, for a robust and reliable pipeline, traditional fine-tuning remains essential. Overall, while foundation models and prompting open promising new directions, fine-tuning coupled with high-quality annotated data continues to represent the most dependable strategy for domain-specific applications.

5.2 Original Contributions and Methodological Adaptations

This chapter builds upon well-established object detection and segmentation architectures, namely YOLO-based detectors and Vision Transformer-based segmentation models. The original contribution of this work does not reside in the development of novel detection or segmentation architectures per se, but rather in their adaptation, integration, and deployment within a unified perception pipeline tailored for autonomous harvesting in a previously unexplored agricultural scenario.

Specifically, pretrained YOLOv5 and YOLOv8 models were employed as baseline detection frameworks and subsequently fine-tuned on custom datasets of edible flowers and blackberries in order to address the domain shift between general-purpose datasets (e.g., COCO) and the target application. In this context, the proposed FLOLO and Berr-YOLO models represent task-specific adaptations of existing architectures through multi-stage transfer learning, partial backbone freezing, and the introduction of targeted data augmentation strategies aimed at improving cross-domain generalization.

In addition, the Segment Anything Model (SAM) was integrated into the percep-

tion pipeline not as a standalone segmentation framework, but as a prompt-driven refinement module and as an automatic annotation tool during dataset generation. A mask-selection strategy based on centroid proximity between predicted bounding boxes and candidate segmentation masks was further introduced in order to ensure consistent instance-level segmentation in cluttered scenes.

5.3 YOLO-Based Object Detection

Being a one-stage detector, YOLO has become the de facto state-of-the-art architecture for real-time object detection. Because of its speed and reliability, it has been adopted across diverse domains, ranging from autonomous driving [120] and manufacturing [121] to agricultural robotics and autonomous harvesting [122].

Since its first introduction by Redmon et al. [67], the YOLO family of models has been continuously updated, modified, and extended. Over time, the architecture has evolved from a conventional object detection algorithm into a versatile framework capable of addressing multiple computer vision tasks, including semantic and instance segmentation, keypoint detection, and pose estimation. Among the most widely adopted distributions is the implementation maintained by Ultralytics (Judicial Wy, MD, USA), a company that actively develops and improves YOLO models, making them freely available for research and educational use under the AGPL-3.0 license. In addition to this widely used release, numerous alternative implementations and adaptations have been developed by independent researchers and companies, targeting specific applications. For instance, some variants optimize bounding box alignment, while others integrate architectural modifications to improve accuracy, inference speed, or task generalization.

Within the extensive YOLO ecosystem, some versions have proven especially impactful, representing pivotal milestones in the model's evolution. YOLOv5 [123] became highly influential thanks to its efficient and accessible PyTorch implementation. More recently, YOLOv8 [124] has further advanced the framework by incorporating object segmentation.

5.3.1 YOLO V1

The core idea behind YOLO is to reformulate object detection as a single regression problem, where a convolutional neural network directly predicts bounding boxes and the associated class probabilities. This unified formulation is what makes YOLO both fast and efficient compared to earlier multi-stage detection pipelines.

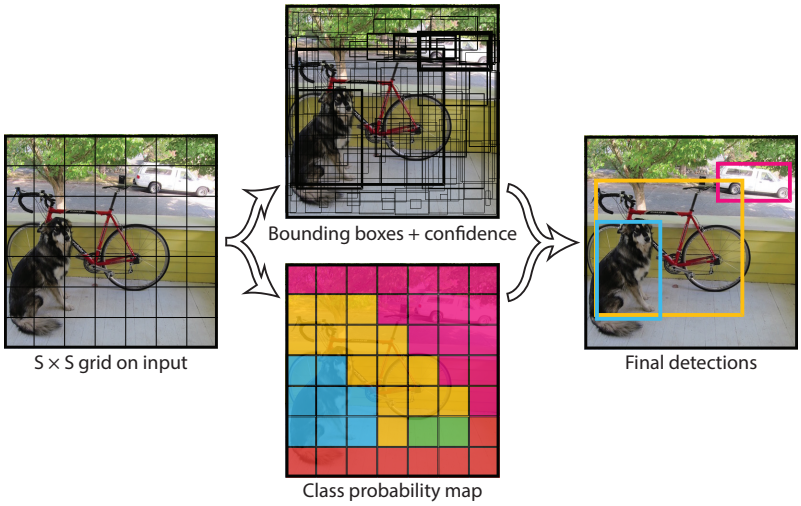


Figure 5.1: YOLO v1 pipeline: the input image is divided into an $S \times S$ grid. Each grid cell predicts B bounding boxes with associated confidence scores and class probabilities. Non-maximum suppression is then applied to eliminate duplicate predictions. Fig. from [67]

As illustrated in Fig. 5.1, the YOLO v1 architecture divides the image into an $S \times S$ grid (S is set to 7). Each grid cell is responsible for predicting B (set to 2) bounding boxes along with a confidence score and C (set to 20) conditional class probabilities, $Pr(\text{Class}_i \mid \text{Object})$. The final class-specific confidence score for each bounding box is computed by multiplying the confidence score by the conditional class probabilities. This encodes both the likelihood of the class being present and the quality of the bounding box fit. To obtain the final detections, a threshold is applied to these scores, and Non-maximum suppression (NMS) is used to remove redundant overlapping boxes, ensuring that only the most confident predictions remain.

The division of the image into grid cells is not applied directly to the input image but instead emerges implicitly from the network architecture. As shown in Fig. 5.2, the convolutional and pooling layers progressively downsample the input until producing a final feature map of size 7×7. These layers, which are responsible for extracting visual features, are commonly referred to as the backbone of the network. The final 7×7 feature map is then interpreted as a grid, where each cell corresponds to a region of the original image. During prediction, each grid cell is responsible for detecting objects whose center lies within its assigned region.

The final fully connected layers output a tensor of size $7 \times 7 \times 30$, corresponding to $7 \times 7 \times (B \cdot 5 + C)$. Here, 5 represents the parameters of each bounding box prediction (x, y, w, h, and confidence), while B is the number of bounding boxes predicted per

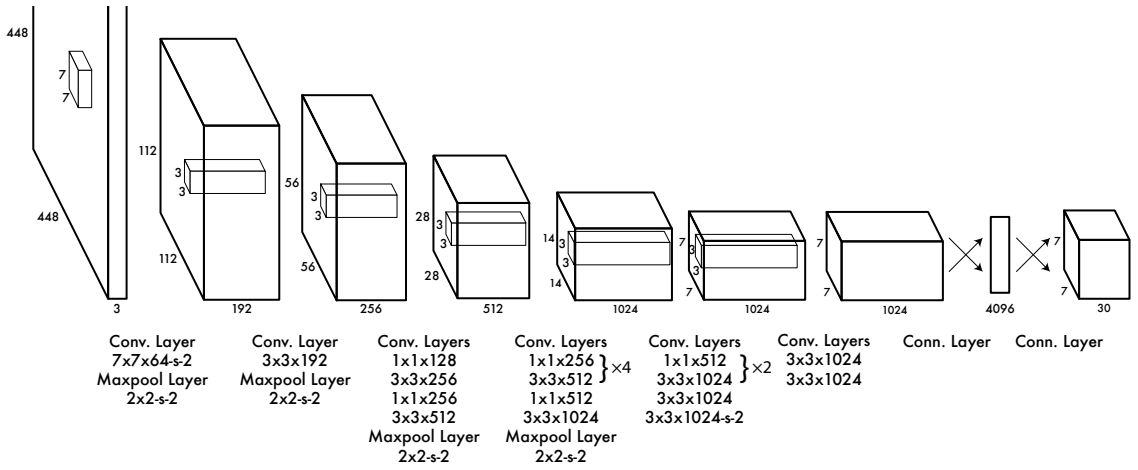


Figure 5.2: YOLO v1 architecture from Redmon et al. [67]. The input image is first resized and processed by a network composed of 24 convolutional layers followed by two fully connected layers. The final prediction is structured as a 7×7 grid, corresponding to the spatial resolution of the last convolutional feature map

grid cell and C is the number of object classes. Adjusting these two hyperparameters (B and C) directly changes the dimensionality of the final prediction tensor.

The design of YOLO v1 makes it exceptionally fast and efficient; however, an intrinsic limitation of this architecture is its difficulty in detecting small or densely clustered objects. This limitation arises from the fact that each grid cell is constrained to predict only B bounding boxes (a tunable hyperparameter, typically set to 2) and to assign them to a single object class. Consequently, when multiple small objects are located close together within the same grid cell, or when an object lies on the border between adjacent cells, the network often produces only a single bounding box, failing to detect each individual instance.

5.3.2 YOLO V5

YOLOv5 was released by Ultralytics in 2020 and quickly became one of the most widely used object detectors. It inherits many of the improvements introduced in YOLOv3 and YOLOv4 [125, 68]. Its rapid adoption was largely driven by the shift from the Darknet framework to a native PyTorch implementation [126], which greatly simplified fine-tuning and deployment

As illustrated in Fig. 5.3, the model combines a modified CSPDarknet53 backbone with an Spatial Pyramid Pooling—Fast (SPPF) block and a CSP-PAN neck, while retaining a YOLOv3-style, anchor-based detection head [127]. The backbone extracts features using Cross-Stage Partial (CSP) connections [128] and incorporates

spatial pyramid pooling to enlarge the receptive field at low cost [129]. The neck then fuses multi-scale features via a PANet-style path aggregation, improving information and localization flow across scales [130]. Finally, the head predicts bounding boxes at multiple feature-map resolutions as offsets relative to predefined anchors; in practice, YOLOv5's AutoAnchor routine estimates dataset-appropriate anchor sizes during training, improving matching and convergence [127].

5.3.3 YOLO V8

One of the main innovations of YOLOv8 is its anchor-free detection strategy. Unlike YOLOv5, which relies on predefined anchor boxes and predicts offsets, YOLOv8 directly estimates the center and dimensions of bounding boxes. This reduces the number of required predictions, simplifies post-processing steps such as NMS, and consequently speeds up inference [131, 132].

As illustrated in Fig. 5.4, the overall architecture of YOLOv8 still follows the standard three-part structure: Backbone, Neck, and Head. The backbone is based on a custom version of CSPDarknet53 with C2f modules, enabling efficient feature extraction [133]. The neck employs a PANet structure to ensure effective multi-scale information flow. The head, which is task-specific, is responsible for generating the final predictions.

In addition to object detection, YOLOv8 supports multiple tasks such as segmentation, pose estimation, and tracking [127]. For object detection, the head resembles that of YOLOv5, but with anchor-free predictions as described above [135]. For other tasks, such as semantic segmentation, an additional mask prediction branch is included. This branch leverages multi-scale feature maps to generate per-instance masks, following a strategy similar to YOLACT [136].

5.4 Segmentation with Vision Transformers

Since the introduction of the Transformer architecture by Vaswani et al. in 2017 [137], encoder–decoder models based on multi-head self-attention have rapidly become one of the most widely adopted approaches in deep learning. The key innovation of this architecture is the ability to attend to different parts of the input simultaneously and capture long-range dependencies, making it a breakthrough initially for Natural Language Processing (NLP).

Although Transformers were originally designed for sequential data such as text, they were soon extended to other domains. In 2020, Dosovitskiy et al. [72] intro-

duced the ViT, adapting the Transformer framework to image processing.

As illustrated in Fig. 5.5, an input image is divided into fixed-size patches, each of which is flattened and enriched with a positional embedding to retain spatial information. These patch embeddings are then fed into the Transformer encoder in the same way that words are processed in a sentence. The encoder relies on multi-head self-attention, where multiple attention modules operate in parallel to learn complementary relationships in the latent space [72]. This design enables the model to capture both local and global dependencies efficiently while maintaining high parallelization during training.

For classification tasks, ViT employs a special [Classify token (CLS)] token, which aggregates information from all patches. After processing through the encoder, the [CLS] token is passed to a classifier to predict the final image label. By leveraging multi-head self-attention, ViT can correlate information across distant patches, achieving competitive performance with CNNs and excelling in a wide range of computer vision tasks [73].

Beyond classification, the Transformer architecture has been adapted for various dense prediction tasks. For example, object detection is addressed with DETR [73], while semantic segmentation has been tackled with models such as the SEgmentation TRansformer (SETR) [138] and SegViT [139]. In SETR, a standard Vision Transformer serves as the encoder backbone to extract feature representations. These features are then upsampled by a decoder, which can be as simple as a naïve decoder that linearly upsamples patch-level features into class logits, or a more sophisticated Progressive Upsampling (PUP) decoder that integrates convolutional layers for gradual refinement.

Similarly, SegViT also uses a plain ViT encoder, but introduces a novel decoder: the Attention-to-Mask (ATM) module. As shown in Fig. 5.6, ATM employs learnable class tokens that interact with the encoded patch features to identify regions with high similarity to specific classes. These similarity maps are then transformed into segmentation masks and combined with class predictions to produce the final semantic segmentation output.

5.4.1 Zero-Shot Segmentation with the SAM Model

The SAM, introduced by Kirillov et al. [33], has quickly become a foundational model for image segmentation. Similar to other foundation models such as DALL · E or GPT-3, SAM can be adapted to a wide variety of tasks using only simple prompts [140]. Its architecture is both straightforward and efficient: the input image is first

processed by a pre-trained ViT to generate an image embedding, which is then passed to a mask decoder, a customized Transformer decoder block. At the same time, the user-provided prompt is encoded by a prompt decoder. A key feature of SAM is its ability to handle multiple forms of prompts, including points, bounding boxes, text, or even masks (see Fig. 5.7).

The significance of SAM lies not only in its performance, but in its zero-shot capabilities. With no task-specific fine-tuning, the model can generate high-quality segmentation masks given only a prompt, or alternatively, it can automatically segment all objects in an image without requiring any prompt at all. This is possible because SAM was trained on SA-1B, the largest segmentation dataset to date, containing over 1 billion masks across 11 million images, which was released together with the model.

Another important property of SAM is its compatibility with object detection models. Since it accepts bounding boxes or points as input prompts, it can be seamlessly integrated with detectors such as YOLO. For example, the bounding box coordinates (or their centers) produced by YOLO can be fed into SAM, which then refines these detections into high-quality segmentation masks—effectively enabling zero-shot instance segmentation.

5.5 Proposed Detection-Segmentation Pipeline

Two different pipelines were tested. The first combined the detection capabilities of YOLOv5 with the segmentation power of SAM. The second relied solely on YOLOv8, while SAM was employed only during the annotation phase.

In both cases, only the YOLO architecture was fine-tuned. Specifically, YOLOv5 was fine-tuned for detecting edible flowers, resulting in the model named *FLOLO*, while YOLOv8 was fine-tuned for detecting and segmenting blackberries, resulting in *Berr-YOLO*. Each YOLO version provides multiple model sizes—nano, small, medium, large, and extra-large—ranging from $1.9M$ to $86.7M$ parameters in YOLOv5 and from $3.2M$ to $68.2M$ in YOLOv8. Larger models generally achieve higher detection accuracy but at the cost of reduced inference speed. For YOLOv5, the large version (YOLOv5l, $46.5M$ parameters) was used, while for YOLOv8, the nano version (YOLOv8n, $2.9M$ parameters) was chosen.

The edible flower detector, *FLOLO*, was developed from YOLOv5l through two consecutive fine-tuning stages. First, the model was initialized with COCO-pretrained YOLOv5l weights and fine-tuned on the D0 dataset, resulting in *D0-FLOLO*. In the second stage, the model was further fine-tuned on the custom Flo-

raDet dataset.

Both training phases were conducted for 300 epochs using standard YOLO hyperparameters: a learning rate of 0.01, momentum of 0.937, and weight decay of 0.005 with the SGD optimizer. The standard YOLO loss function, comprising localization, confidence, and classification losses, was employed. For each phase, the datasets were split into 75% training, 15% testing, and 10% validation sets. This procedure was applied to images from both data collection campaigns; consequently, the images collected in November and July were combined and divided according to the same proportions(75%, 15%, and 10%).

The final models were selected based on the epoch achieving the best validation performance. A summary of the architectures, training sets, and results is provided in Table 5.1.

Table 5.1: Details on *D0-FLOLO* and *FLOLO* architectures. Reported are the main performance metrics, starting weights, and training sets for the two fine-tuned models.

Model Name	**Training Set**	**Starting Weights**	**Best Epochs**	**mAP@0.5**	**Precision**	**Recall**	**Det. Val. Error**
D0-FLOLO	D0	YOLOv5-large	284	0.97	0.96	0.93	0.0039
FLOLO	FloraDet	D0-FLOLO	229	0.68	0.67	0.68	0.045

For the blackberry detection task, *Berr-YOLO* was carefully fine-tuned to ensure strong cross-domain adaptation, as the training dataset included out-of-domain images. To achieve this, several strategies were employed. First, partial freezing of the backbone was applied: the first eight layers were frozen to preserve the pretrained feature extraction capability. Second, a set of data augmentation techniques was introduced, including mosaic augmentation, scaling, flipping, and rotation, to improve generalization. The datasets used for training and evaluation, along with image and instance counts, are summarized in Table 5.2.

Table 5.2: Datasets used for training and evaluation. The table reports the division between training, test, and validation sets, together with the number of images and annotated instances for DF1 and DF2.

	Images		Instances	
Division	DF1	DF2	DF1	DF2
Train	79	42	495	185
Test	7	2	54	7
Validation	8	2	47	9

5.6 Results

This section presents both qualitative and quantitative evaluations of the detection-segmentation pipeline on the edible flower and blackberry datasets. Metrics such as precision, recall, IoU, and inference speed are discussed alongside visual examples.

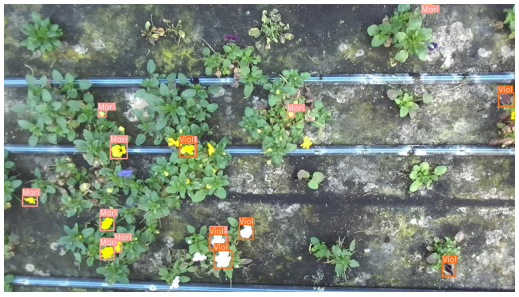
5.6.1 Detection and Segmentation of edible flowers

Overall, the three metrics (mAP, precision, and recall, Table 5.1), demonstrate that FLOLO performs well in detecting flowers, achieving a strong balance between precision and recall. Specifically, the model's output on test images highlights its effectiveness in detection but reveals occasional challenges in classifying pansies (Figure 5.8d). This difficulty may be attributed to the underrepresentation of pansies in the dataset, as well as the natural variability within the species, which includes two distinct types and varieties. It is also important to note that the key performance metric, the object detection error on the validation set, scored a low value of 0.045, indicating the model's success in accurately identifying and localizing ripe flowers. A closer look at the detection performance is provided in Figure 5.9 where some close up detection samples are provided.

(a) FLOLO output for snapdragon

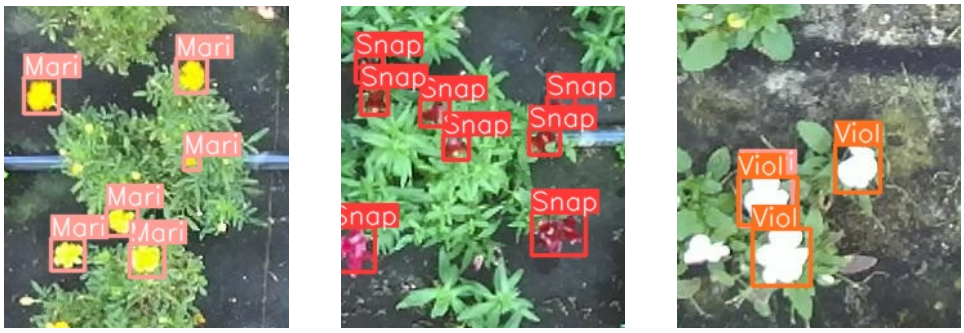
(b) FLOLO output for marigold

(c) FLOLO output for pansy



(d) FLOLO output for pansy

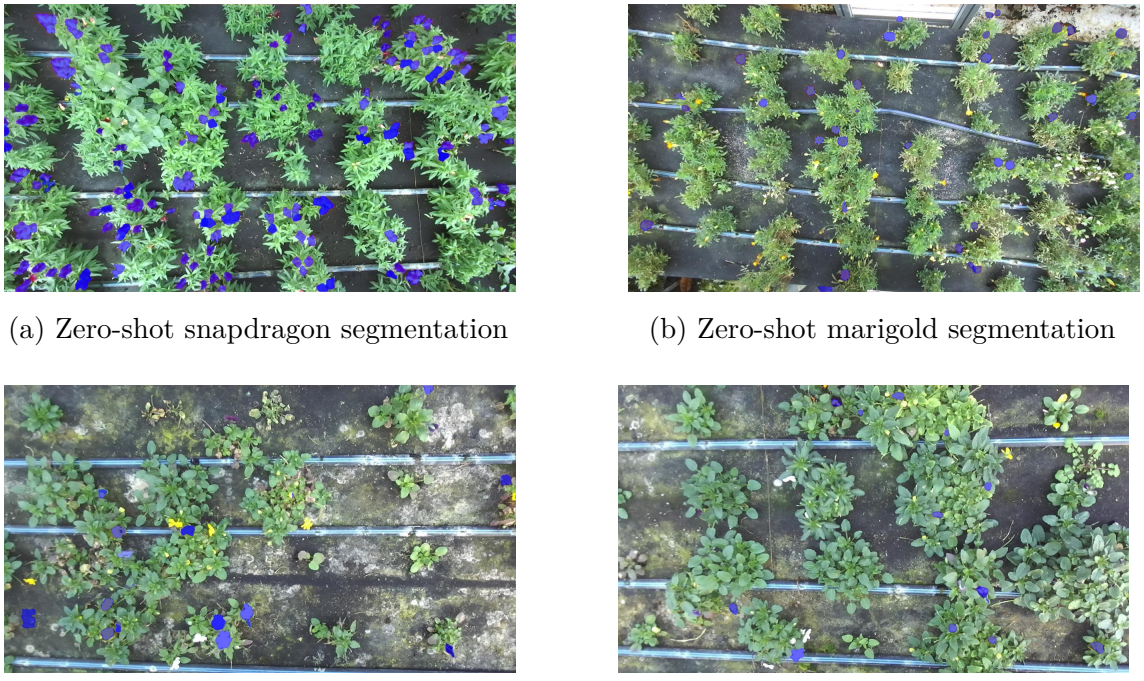
Figure 5.8: FLOLO outputs for different flowers: (**a**) snapdragon, (**b**) marigold, (**c**) viola (1st view), (**d**) viola (2nd view).



(a) Marigold (b) Snapdragon (c) Pansy

Figure 5.9: Cropped close-up output from the preceding images (Mari = marigold, Snap = snapdragon). Not-ready-to-be-picked flowers are correctly not detected.

For the segmentation setup, the bounding boxes predicted by YOLO were used as prompts for SAM. However, in some cases, SAM produced multiple candidate masks for a single bounding box. To address this, the centroid of each polygonal mask and the centroid of the corresponding bounding box were computed, and the mask whose centroid was closest to that of the bounding box was selected. An example of flower segmentation is shown in Fig. 5.10.



(a) Zero-shot snapdragon segmentation (b) Zero-shot marigold segmentation

(c) Zero-shot pansy segmentation (d) Zero-shot pansy segmentation

Figure 5.10: SAM zero-shot segmentation for different flowers: (**a**) snapdragon, (**b**) marigold, (**c**) pansy (1st view), (**d**) pansy (2nd view).

While this work is primarily a proof of principle, it is interesting to examine the speed performance of the vision module for its prospective use as an AI tool for on-field application. In Table 5.3 ir reported the average time, in seconds, required to process the validation images with a varying number of flowers for two varieties of interest, for a total of more than 1400 flowers.

Table 5.3: Average inference time (seconds) for each step of the vision framework. Performed on an NVIDIA GeForce RTX 3090. Averages are computed from 10 images per flower type, covering 457 marigold and 975 snapdragon flowers.

Flower Variety	**Model Performance**		**Time of Task (ToT)**	
	Detection - YOLO	**Segmentation - SAM**	**Per Image**	**Per Flower**
Marigold	0.285	0.551	0.837	0.018
Snapdragon	0.293	0.558	0.851	0.0087

The performance results indicate that the system is well-suited for real-time picking applications. In practice, neither the robotic manipulator nor the subsequent pipeline stages would benefit significantly from further increases in detection or segmentation speed. Moreover, the data are extremely robust, showing an almost constant elapsed time, with a variance on the order of 1×10^{-5}. For example, in the segmentation task, processing the first image takes, on average, about 2.5 times longer than subsequent images, which consistently require around 0.47 seconds each. This difference occurs because the model weights are loaded during the processing of the first image.

As demonstrated in the work of Subramanian and colleagues [141], where a robotic saffron flower harvesting system was developed, approximately one second is allocated for image processing, while the overall task is completed in about six seconds. It is also evident that the total processing time is not directly proportional to the number of flowers; rather, it is dominated by the time required to process each individual image or frame. By adopting a top-down camera view, the system maximizes the number of flowers captured per image, thereby improving the efficiency of detection and segmentation.

5.6.2 Detection and Segmentation of blackberries

The fine-tuned YOLO model achieved an mAP50 of 0.87, demonstrating strong object detection performance. Precision and recall were 0.86 and 0.78, respectively, indicating a well-balanced model with a slight preference toward precision.

The model also showed promising results in inference under out-of-domain conditions. It was able to correctly detect blackberries both with the high-resolution

RealSense camera and with the USB endoscopic camera embedded in the gripper palm. Tests were conducted on a synthetic blackberry bush in controlled environments and in real field scenarios, where large variations in brightness were present.

Examples of detection and segmentation are shown in Fig.5.11. Subfigures 5.11a, 5.11b, and 5.11c illustrate detection and segmentation of blackberries in both controlled and real-world environments. The blackberries appear red due to the overlaid segmentation mask, which closely adheres to the actual fruit contours. Subfigures 5.11d and 5.11e present examples where only bounding boxes are plotted. Occasionally, the model produced false positives, as shown in 5.11e, likely due to the domain gap between training data and deployment conditions.

In terms of processing speed, the vision system operated at approximately 30 frames per second, corresponding to a per-frame processing time of about 0.033 seconds. Since the evaluation was conducted under conditions with only a single blackberry present per frame, the per-instance processing time was equivalent. Notably, in this application the model was executed on the CPU, specifically an 11th Gen Intel(R) Core(TM) i7-1165G7 running at 2.80 GHz.

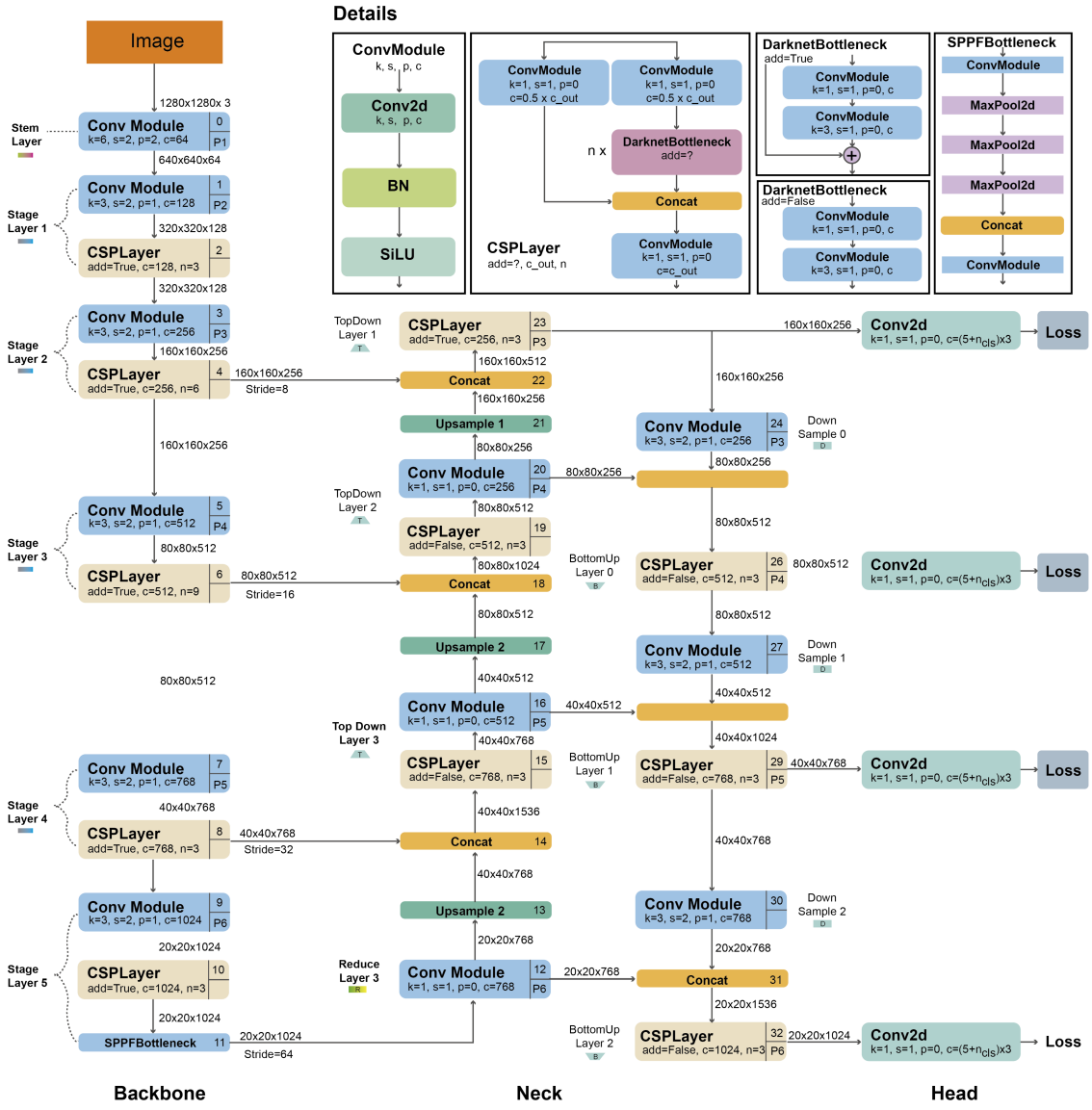


Figure 5.3: YOLOv5 architecture showing the backbone, neck, and prediction heads. Figure from Terven et al. [127]

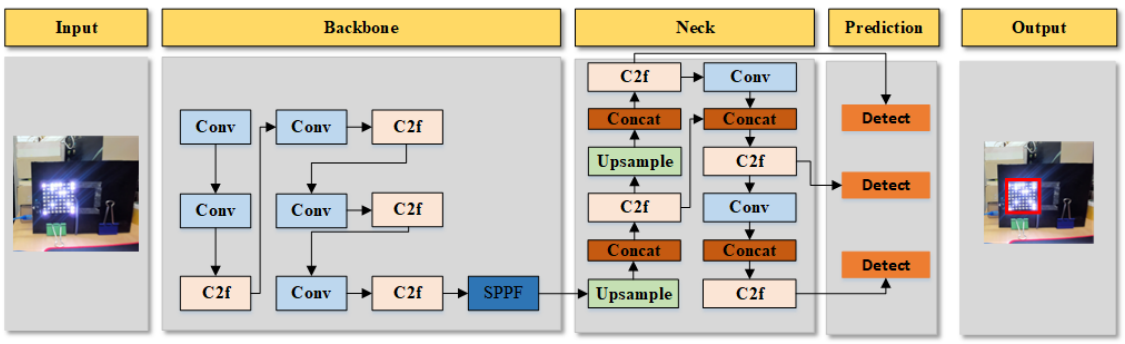


Figure 5.4: YOLOv8 schematic architecture with task specific detection head. Figure from Herfandi et al. [134]

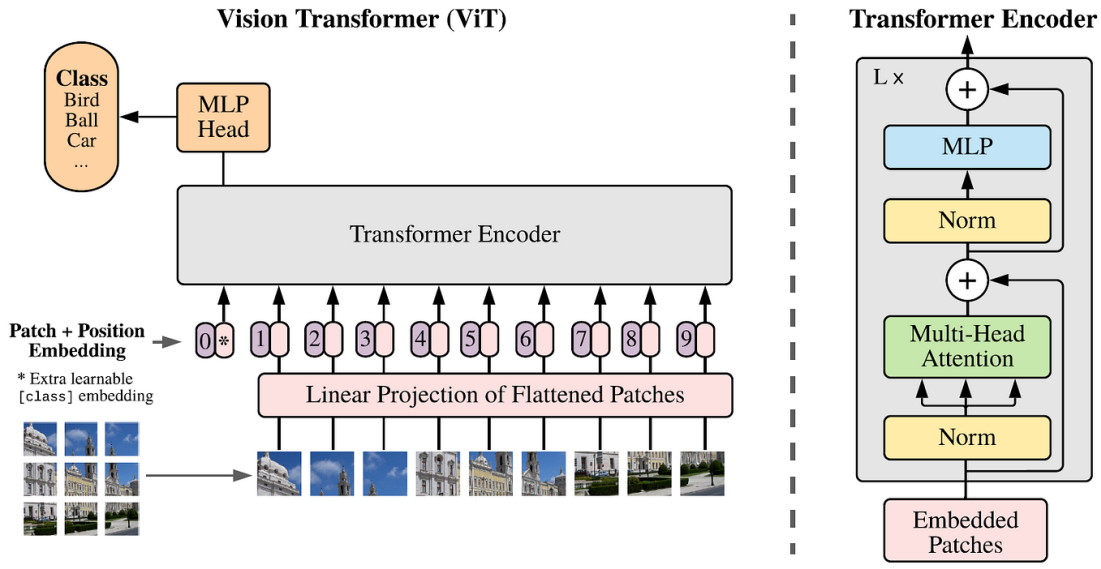


Figure 5.5: Schematic overview of the ViT. An image is divided into patches, flattened, and combined with positional embeddings before being processed by a Transformer encoder. The [CLS] token aggregates global information for image classification. Figure from Dosovitskiy et al. [72]

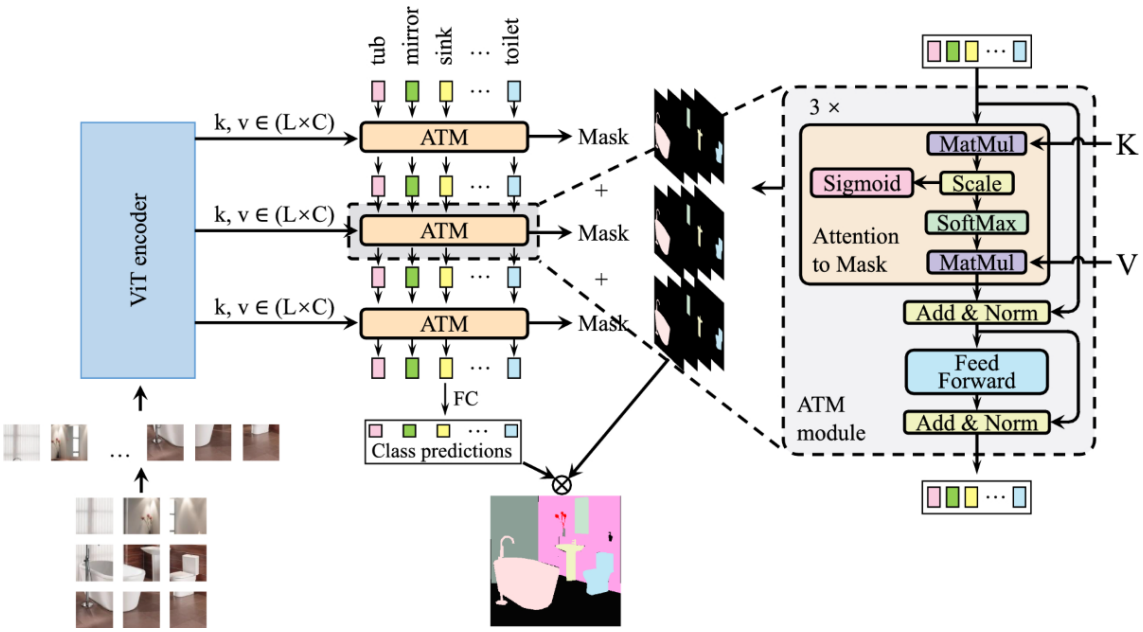


Figure 5.6: Overview of SegViT. A plain ViT encoder extracts patch features, which are passed to the proposed Attention-to-Mask ATM modules. The ATM uses class tokens to directly generate class-specific masks from attention maps, while also updating the tokens for class prediction. Outputs from multiple ATM layers are combined to produce the final segmentation map. Figure from Zhang et al. [139]

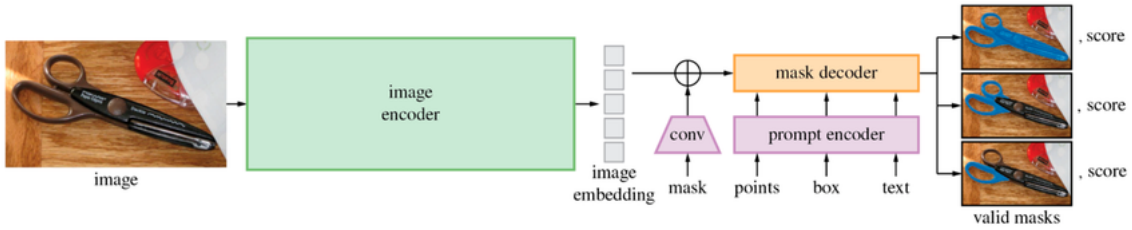
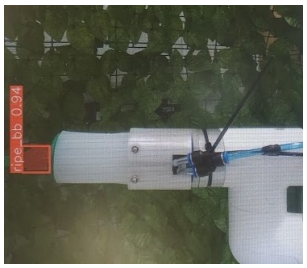
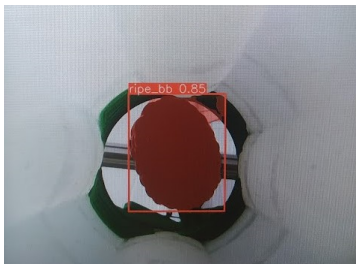


Figure 5.7: Overview of the SAM. A ViT encoder generates image embeddings, which are combined with prompts (points, boxes, text, or masks) via a prompt decoder. The mask decoder then produces segmentation masks at near real-time speed. Figure from Kiriloc et al. [33]

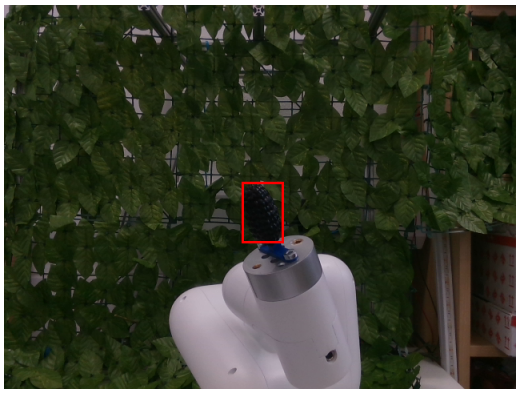


(a) Synthetic blackberry detection (RealSense).

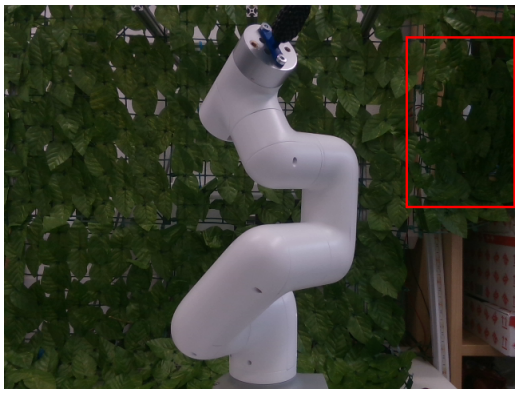


(b) Detection from endoscopic camera.

(c) Detection from endoscopic camera.



(d) Synthetic blackberry detection.



(e) Erroneous detection (false positive).

Figure 5.11: Performance of the fine-tuned YOLOv8 model. (**a**) Detection on synthetic blackberries using both the RealSense and endoscopic cameras. (**a**) Detection and segmentation of blackberries in real field conditions. (**a**) Detection on synthetic blackberries in a controlled setup. (**a**) Example of erroneous detection (false positive) due to out-of-domain training data data.

Chapter 6

3D Pose and Plucking Point Estimation

6.1 Problem Statement

To accurately describe an object in three-dimensional space, both its position (a point) and orientation must be specified [142]. For perfectly symmetric objects, such as a sphere in Euclidean space, orientation is irrelevant. In agricultural robotics, however, even highly symmetric fruits such as oranges possess a stalk, making it necessary to represent the fruit with both a keypoint and an orientation [15, 143]. The keypoint is commonly referred to as the plucking point, since many crops (e.g., strawberries, peppers, and cucumbers) must be detached by cutting or severing the stalk [98, 97]. Orientation, on the other hand, is described through 3D pose estimation, typically represented by a 3×3 rotation matrix that defines the principal axis along which the fruit develops.

Because robotic harvesting requires direct physical interaction with the target object, both the plucking point and pose estimation are essential for implementing an effective picking strategy [144]. In practice, fruits are usually detected first in the 2D domain of an RGB image, after which precise localization and orientation estimation are needed to guide the manipulator for a successful harvest.

Since robotic systems must perceive and operate in 3D space, their perception modules often rely on sensors that extend beyond the 2D domain. For example, stereo cameras can provide depth maps or point clouds, while LiDAR sensors are also widely used. This allows the vision module to combine RGB images with 3D data of the target object. Furthermore, segmentation masks obtained from RGB images can be projected into the 3D domain to isolate the corresponding points in

a point cloud. In the case of ZED cameras, where the 3D point cloud is aligned with the left image, object points can be selectively extracted by applying the 2D segmentation mask directly. Similarly, with Intel RealSense cameras, the 2D mask can be mapped onto the 3D point cloud using the intrinsic calibration parameters of the device.

After this projection step, a segmented 3D point cloud containing only the points belonging to the object of interest can be isolated from the background. Pose estimation then consists of inferring from this point cloud a transformation that represents the object's orientation. While, many approaches rely on 3D point clouds for this purpose, only a few attempt to infer orientation directly from 2D images. For example, Le Louëdec et al. [145] proposed a method for strawberries in which orientation is estimated directly from RGB images by detecting two characteristic keypoints and computing their relative positions.

6.2 Original Contributions and Methodological Adaptations

This chapter builds upon established methods for 3D pose estimation and plucking point determination, including PCA, geometric model fitting, and keypoint-based orientation estimation. Its main contribution lies in the adaptation and integration of these techniques within a unified perception pipeline for autonomous harvesting of edible flowers and blackberries.

For flowers, a novel regression-based model was introduced to estimate the optimal plucking point. This model was derived from a custom dataset of flower measurements and validated through consultation with plant science experts, ensuring that the predicted plucking points are biologically consistent.

For blackberries, the primary novelty resides in the validation of pose estimation. The dataset and ground-truth were generated using the precise coordinates and orientations provided by the robotic manipulator's kinematics, allowing fully autonomous and accurate labeling. This approach represents a largely unexplored method for generating reliable pose annotations in agricultural robotics.

Together, these contributions enable robust, precise, and biologically informed pose and plucking point estimation, demonstrating practical applicability under real-world harvesting conditions.

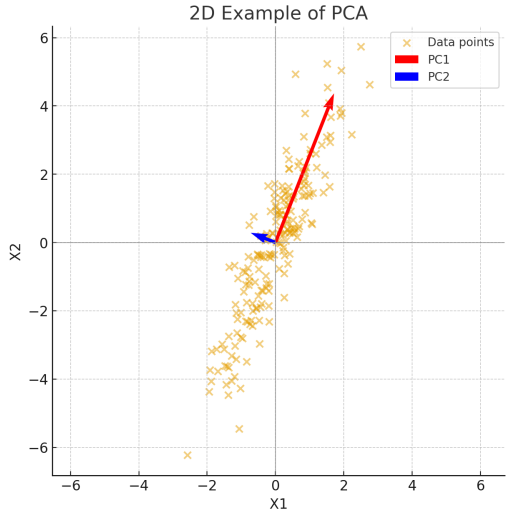
6.3 PCA-Based Pose Estimation

One of the most widely used methods for fruit pose estimation from point clouds is PCA. The popularity of PCA lies in its simplicity, unsupervised nature, and effectiveness in extracting the dominant geometric structure of 3D objects. By analyzing the variance in the distribution of points, PCA identifies orthogonal directions that represent the main axes of the object of interest. When applied to an isolated fruit point cloud, the first principal component often aligns with the natural growth axis of the fruit, thereby providing a straightforward estimate of its orientation.

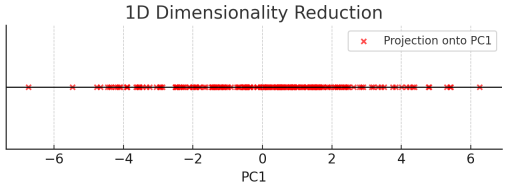
6.3.1 Principal Component Analysis

PCA is an unsupervised statistical and machine learning technique that does not require annotated data. It is primarily used for dimensionality reduction and multivariate data analysis, allowing to show latent linear relationships between sets of variables [146]. The fundamental objective of PCA is to identify a reduced set of orthogonal components that retain as much of the original variance in the dataset as possible, thereby maximizing the explanatory power with fewer variables [147].

Formally, given a dataset represented by a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, where n is the number of observations and p the number of variables, PCA seeks a set of orthogonal directions (principal components) $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k$ such that the variance of the projected data $\mathbf{X}\mathbf{u}_i$ is maximized subject to orthogonality constraints. These directions correspond to the eigenvectors of the covariance matrix $\mathbf{S} = \frac{1}{n-1}\mathbf{X}^\top\mathbf{X}$ while their associated eigenvalues indicate the proportion of total variance explained by each component.



(a) Two-dimensional illustration of PCA



(b) Dimensionality reduction from 2D to 1D

Figure 6.1: Illustration of PCA. (a) Data points (gray) with the two principal components: PC1 (red), capturing the maximum variance, and PC2 (blue), capturing the remaining variance. (b) Dimensionality reduction from 2D to 1D, where all data points are projected onto the PC1 axis, yielding a one-dimensional representation of the dataset.

An illustrative example of PCA in a two-dimensional domain is shown in Fig.6.1. As depicted in Subfig.6.1a, the algorithm identifies two orthogonal components that maximize the explainable variance of the dataset. In the specific case where the dimensionality is reduced from 2D to 1D, the data are projected onto the principal component that captures the greatest variability—namely, the first principal component—since it retains the most informative features of the dataset (Subfig. 6.1b).

6.3.2 PCA Pose Estimation

Since PCA is an unsupervised technique, it offers a particularly useful property: there is no need for data annotation. This means the method can be directly integrated into a pipeline without requiring any prior training. For this reason, many robotic harvesting systems adopt PCA for crop pose estimation. Starting from an isolated point cloud, the eigenvectors extracted from PCA can be used to define the pose of the target fruit. By ranking these eigenvectors according to their eigenvalues (i.e., explained variance), it is possible to identify the principal fruit axis, which in most cases corresponds to the eigenvector with the largest eigenvalue.

For example, Hussain et al. [148] applied PCA to estimate the orientation of

small green apples for thinning operations. They reported that approximately 65.5% and 61.9% of the inferred orientations were accurate within 15°. Similarly, Li et al. [81] employed PCA to estimate the pose and plucking points of high-quality tea leaves. As shown in Fig.6.2, given an isolated point cloud, PCA generates three direction vectors (in red and blue). Among these, the red vector was found to represent the natural growing direction of tea shoots, which also indicates the optimal plucking point. These studies highlight the versatility of PCA and its applicability across different agricultural domains.

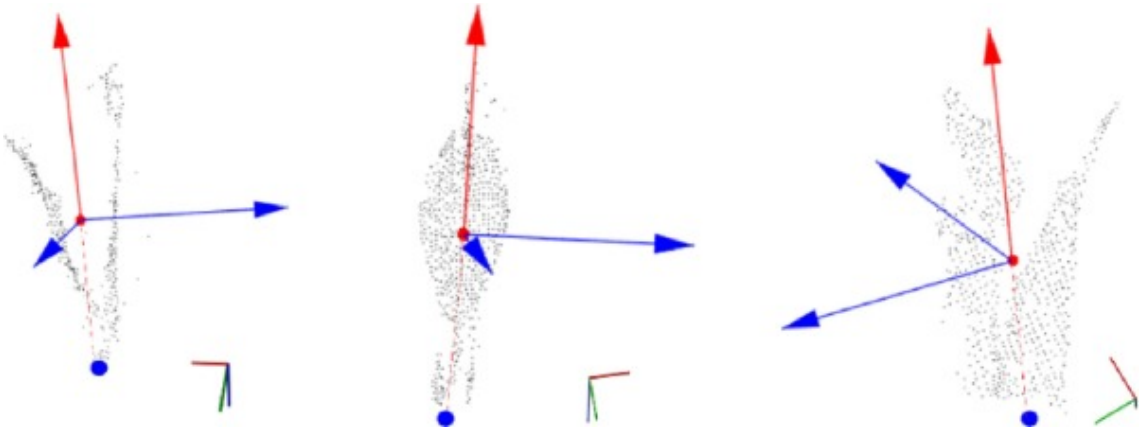


Figure 6.2: Isolated point cloud of tea shoots ready for harvesting, with PCA-derived pose estimation vectors shown in blue and red. The red vector corresponds to the natural growing direction of the shoot, which also indicates the plucking point. Figure from LI et al. [81]

Despite its advantages, PCA also has important limitations. First, the algorithm is agnostic to the morphology of the target point cloud. For example, in the case of fruits, PCA cannot inherently distinguish the top (where the stalk is located) from the bottom. While this limitation is sometimes alleviated by the natural tendency of fruits to grow downward, such an assumption imposes a strong constraint on the method's applicability.

A second limitation arises from the variance-based nature of PCA. When the point cloud has a spherical or nearly spherical shape, variability is uniformly distributed across directions, which leads to unstable or random eigenvectors. This can be diagnosed by inspecting the eigenvalues: if they are very similar, each principal component explains a comparable portion of variance, indicating high symmetry. Indeed, Hussain et al. [148] reported that PCA was applicable to small green apples precisely because their shape is ellipsoidal, which mitigates the issue. Nevertheless, they also introduced a post-processing technique to correct the orientation, addressing the problem of randomness in the PCA direction.

Consequently, PCA is most effective when applied to fruits or objects with a clear dominant axis of growth—that is, those that exhibit a certain degree of asymmetry and are not heavily occluded by leaves or neighboring objects. In contrast, strong occlusions can make the partially observed point cloud appear symmetric, leading to unreliable or ambiguous pose estimates.

6.4 Geometric Approximation via Model Fitting

Beyond indirect methods such as PCA, another class of approaches tackles the problem through geometric fitting. The overall pipeline resembles the one described earlier up to the stage of point cloud isolation. Afterward, however, the algorithm fits a specific geometric figure to the segmented point cloud. In particular, once the general shape of a fruit is identified, the model estimates the parameters of the corresponding geometric figure that best describes the point cloud. For example, Jang et al. [85] estimate the orientation and center of tomatoes by fitting a sphere to the isolated point cloud. The sphere parameters are obtained via a least-squares minimization:

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2 \tag{6.1}$$

Here, the tomato's 3D position is given by the sphere's center point (x_0, y_0, z_0), while its orientation is defined as the vector connecting the previously detected sepal to the sphere center.

A related strategy is presented by Guo et al. [79], who first constructed an offline dataset of target fruits scanned in high resolution with a 3D scanner. Feature-based matching algorithms are then used to register these offline templates to the isolated online point clouds, enabling estimation of fruit pose and orientation.

The main advantage of such approaches is their reliability and precision in localization and orientation estimation, as demonstrated by geometric fitting studies where simple parametric models such as spheres or ellipsoids achieve accurate pose and sizing in controlled conditions [149, 150]. However, challenges arise under occlusion and, more importantly, due to fruit shape variability. For instance, Li et al. [151] show that occluded fruit significantly degrades localization accuracy, even when geometric fitting is applied, while clustered fruits introduce additional errors in ellipsoid fitting [150]. The parametric fitting method is also limited in generalization, since the underlying geometric model must be redefined whenever fruit morphology changes or when the approach is transferred to another crop domain [149]. Template-

based methods are even less generalizable, as highlighted by Nguyen et al. [152], because they require constructing dedicated offline datasets of high-resolution 3D scans of representative fruit samples and typically fail to generalize to unseen shapes or occluded targets.

6.5 Pose Estimation via Keypoint detection

Estimating object pose through keypoint detection is a well-established strategy, particularly in domains such as human pose estimation. In the context of 3D pose estimation, two main approaches are commonly adopted. The first detects keypoints in a 2D image and subsequently maps them onto a 3D point cloud. The second directly detects and outputs 3D keypoints from the input. The latter are known as single-stage methods because they directly regress the 3D pose, whereas the former are referred to as 2D-to-3D lifting methods, since they first infer 2D keypoints and then lift them into 3D space.

Both approaches face important challenges, primarily due to visual ambiguities (particularly when relying solely on 2D images) and, even more critically, due to data scarcity and the limited availability of ground-truth 3D annotations [153]. While these difficulties already affect human pose estimation—a domain that is relatively mature and well-studied—they are even more pronounced in the case of fruit pose estimation, where fewer datasets and specialized solutions exist.

For this reason, keypoint-based methods have not yet become widespread in agriculture. An exception is the work of Shi et al. [154], who applied this approach to peppers. Their method first employs an object detection model to localize the peppers. The detected regions containing the peduncle are then passed to a dedicated keypoint detection stage, where three keypoints—top, middle, and bottom—are extracted using a newly developed Lite Vision Transformer (LVT). These keypoints are used to infer the orientation of the peduncle skeleton. However, the study does not provide full 3D pose estimation. The authors note that an RGB-D camera could be used to extend the 2D skeleton into 3D, but this was not experimentally validated.

The main limitations of this approach mirror those of the general field: sensitivity to occlusion, the gap between 2D and 3D representations, and above all the requirement for large, high-quality annotated datasets. In addition, there is a clear computational trade-off. Keypoint detection networks add significant overhead, and widely used object detectors such as YOLO do not natively output both keypoints and segmentation masks. This means that either the architecture must be redesigned or multiple algorithms (detection and keypoint estimation) must be

run in parallel, which increases complexity and resource demands.

6.6 Proposed Pose and Plucking point estimation

6.6.1 Edible Flowers

The pose estimation module operates sequentially after the detection and segmentation stage. Flowers in the input image are first segmented using the SAM deep learning model, and for each flower of interest, the corresponding binary segmentation mask is used to extract its 3D point cloud. Principal Component Analysis (PCA) is then applied to the point cloud to determine the principal axes that describe its spatial distribution. The flower pose is derived from these axes (Fig. 6.3). Since PCA does not assign a fixed orientation to its components, the axis-direction ambiguity is resolved by designating the first principal component as the growth axis and constraining it to point upwards, consistent with the natural growth direction of most flowers. Due to the high flower density, overlaps may occur; in such cases, the pose is estimated only for flowers with a complete point cloud. As discussed in the introduction, flowers are harvested daily, or even multiple times per day, and in random order: once the upper flower is picked, the flower beneath it becomes exposed and ready for harvesting during the next cycle.

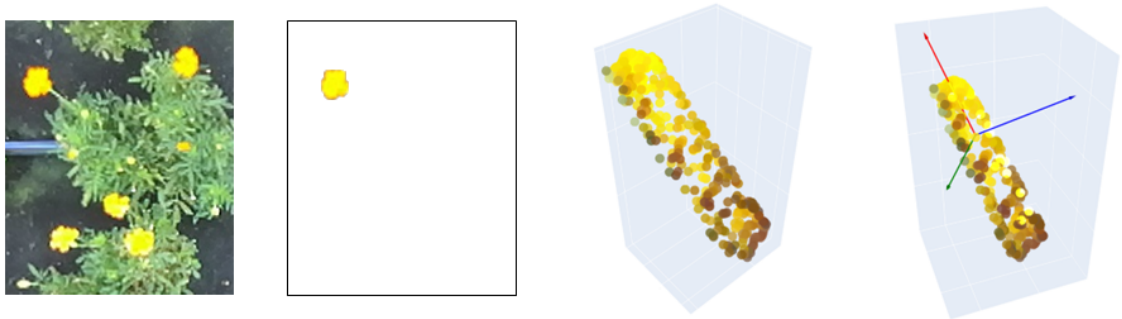
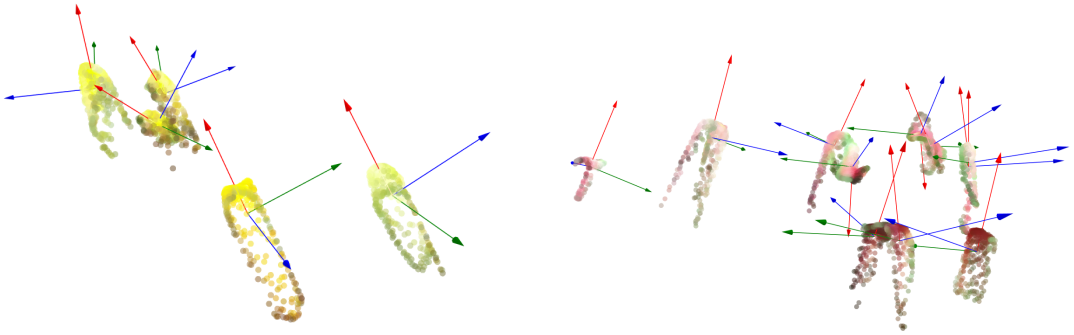


Figure 6.3: Flower pose estimation for steps. From left to right: input image, single flower cutout trough SAM, derived isolated point cloud in the 3D space, and PCA-based point cloud analysis.

An example of the process applied to estimating the pose for each flower to be subsequently utilized to determine the optimal plucking point is shown in Figure 6.4. Pose estimation examples are presented for smaller frames of marigold and snapdragon flowers, previously displayed in Fig. 5.8.



(a) Marigold (b) Snapdragon

Figure 6.4: Isolated point clouds of (**a**) marigold and (**b**) snapdragon flowers and the perspectives vector component. For a closer view, see 6.7, which also shows the estimated plucking points for each flower.

6.6.2 Flower Plucking Point Estimation

Accurately estimating the growth direction of flowers is a necessary step but not sufficient for enabling automated harvesting. A dedicated module is therefore required to determine the optimal plucking point. In manual harvesting, flowers are usually plucked just below the calyx. However, locating this point through pattern recognition is challenging, as it is often occluded by the flower itself or by surrounding leaves. Similarly, 3D shape matching is impractical due to the highly variable and complex morphology of flowers, which contrasts with the consistent shapes typically required for such methods. A schematic representation of the optimal plucking point is shown in Fig. 6.5a.

(a) Side view of a marigold flower featuring pose vectors and the plucking point.

(b) Top view of a marigold flower, showcasing its circumference and diameter (d).

(c) Side view of a marigold flower, highlighting height and plucking point (h).

Figure 6.5: Marigold flowers: side view with pose vectors, top view with circumference and diameter, and height perspective.

The optimal plucking point generally aligns with the extension of the flower's vertical axis (Fig. 6.5a). Given that the flower pose is defined by the three position vectors obtained earlier, the main challenge lies in determining the correct distance from the flower's apex (measured with ZED RGB-D cameras) down to the plucking point. This distance, denoted as h (Fig. 6.5c), has not been explicitly addressed in previous studies. To tackle this problem, we propose estimating h by establishing a numerical relationship between the flower diameter d (Fig. 6.5b) and the distance from the apex to the plucking point (Fig. 6.5c).

Table 6.1: Linear regression equations and corresponding upper boundaries for each flower type.

	Pansy	**Snapdragon**	**Marigold**
Linear regression	$h_v = 0.36d_v + 26.33$	$h_s = 0.38d_s + 5.33$	$h_m = 0.66d_m + 7.10$
Upper boundary	$h_v = 0.36d_v + 43.20$	$h_s = 0.38d_s + 34.57$	$h_m = 0.66d_m + 37.91$

The relationship between h and d was evaluated for each flower variety using linear regression (Fig. 6.6), implemented in Python with the scikit-learn 1.3.2 package [155]. To validate the model, 80% of the data was used for training and the remaining 20% for testing. Although the regression line generally captures the data trend, many observations fall directly on the line, which could result in underestimating

h and potentially damaging flowers during cutting. To address this, the plucking point estimation module does not rely on the regression line itself but instead uses a translated line representing the upper boundary of the 85% confidence interval. Table 6.1 reports both the regression equations and the upper-bound equations for each flower type. The 85% threshold was selected as a compromise between reducing the risk of cutting too shallow and avoiding unnecessary damage. In the test set, underestimation occurred in only one snapdragon flower, one pansy flower, and two marigold flowers. Further field trials will be necessary to fine-tune this threshold.

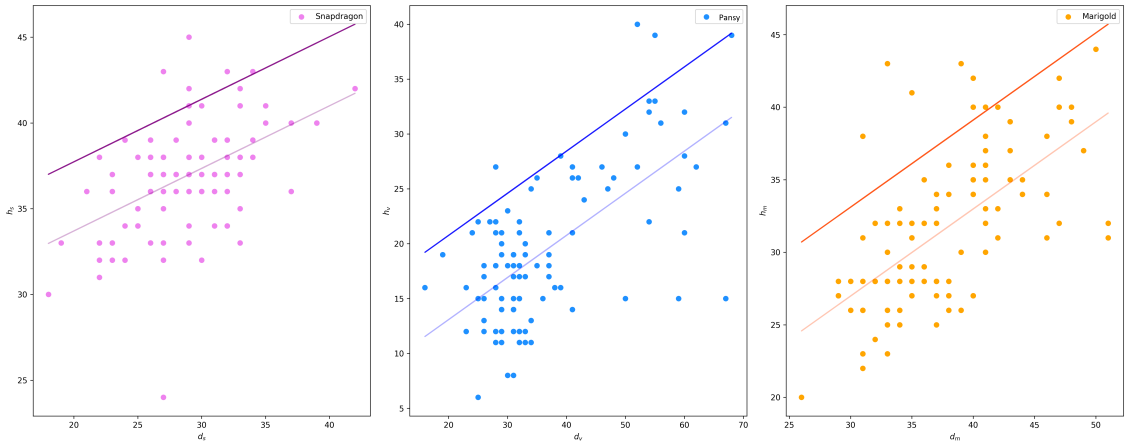
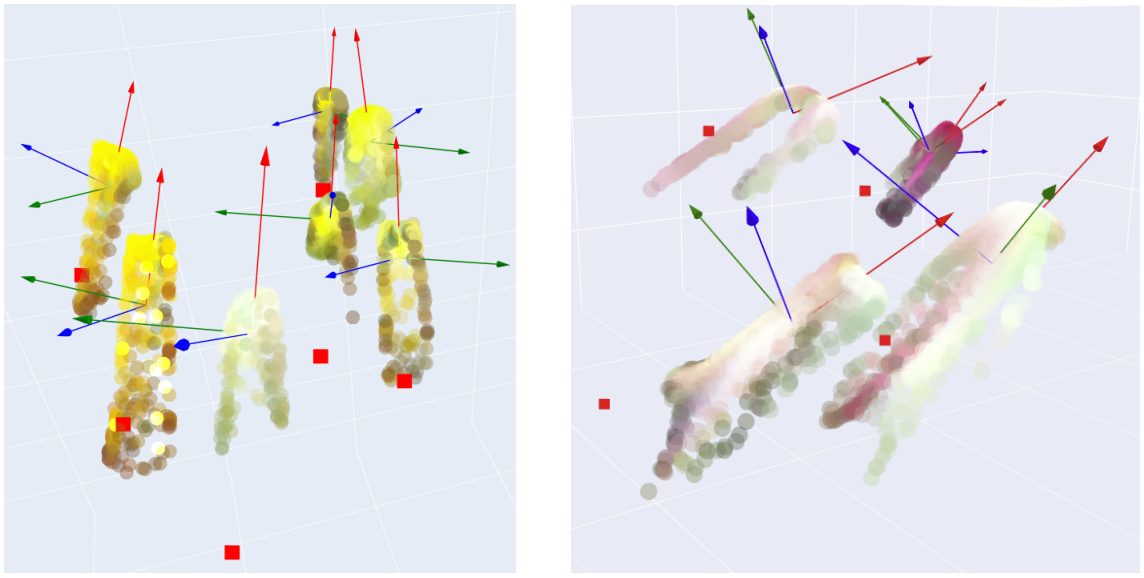


Figure 6.6: Scatter plot of flower diameter (Subfig. **??**b) versus total flower height (Subfig. **??**c). Light solid lines indicate the linear regression fit, while darker lines represent the 85% upper boundary.

In the inference phase, the framework processes the binary mask, the flower point cloud, the pose estimation, and the regression equations reported in Table 6.1. For different flower species, a separate regression model can be derived if needed. The flower center is identified among the uppermost points of the 3D cloud. Using the binary mask, the flower diameter is determined by computing the smallest convex polygon enclosing the point cloud. The center point is then defined as the polygon's interior point with the maximum distance from its boundary, yielding the value of d. Next, the flower height h is estimated using the regression functions. With the pose determined, the center point is translated downward along the main vertical axis by the distance h. This translated point represents the estimated optimal plucking point, highlighted by the red square in Fig. 6.7a. The figure also illustrates the overall framework applied to marigold and snapdragon flowers. The pose vectors are displayed in red, green, and blue, with the red vector denoting the primary axis that describes the flower's orientation. A comparison of the two examples shows that the method accurately estimates the pose for both upright flowers (marigold) and tilted flowers (snapdragon).



(a) Marigold (b) Snapdragon

Figure 6.7: Isolated point clouds of (**a**) marigold and (**b**) snapdragon flowers and the respective vector components (red, green and blue vectors) and estimated plucking point (red squares).

6.6.3 Blackberry Pose Estimation

The localization and pose estimation of blackberries follow a similar procedure. The specific point cloud is first isolated from the data provided by the RealSense depth sensor and then post-processed to improve accuracy. Outlier points are removed using the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm, after which an erosion step eliminates points near the edges to reduce interference from objects in contact with the berry. Finally, the point cloud is refined through downsampling with a 2 mm voxel grid. These preprocessing steps are necessary to mitigate sensor noise, a process that is less critical when using ZED cameras, as their point clouds are already denoised.

The refined point cloud is then used to estimate the berry's pose through PCA. In this case, the orientation along the principal axis is determined through a weighted decision. The preferred orientation is downward; however, if the primary eigenvector points sideways and explains most of the variance, the chosen direction is instead based on the orientation of the second eigenvector.

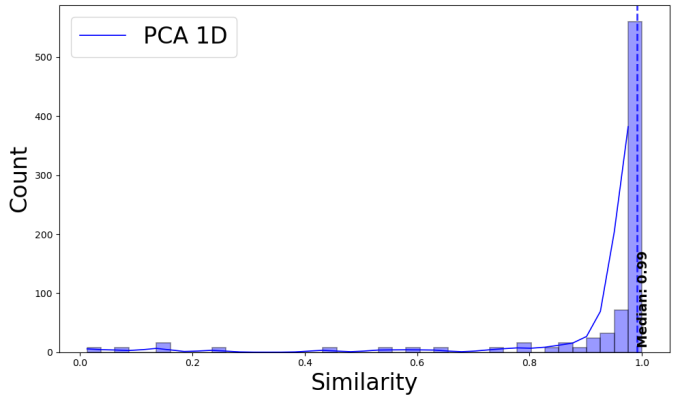


Figure 6.8: Distribution of the absolute dot product between the ground truth main axis and the PCA-inferred axis. The median value is marked with a dashed line.

Obtaining ground truth values for the quantitative evaluation of localization and pose estimation is challenging. Previous works [144, 80, 81] report only qualitative results, typically by overlaying the reconstructed pose onto the real scenario. In contrast, this study provides a quantitative assessment based on 110 tests.

A synthetic blackberry was randomly positioned within the workspace while being held at the tip of a robotic arm, allowing precise ground truth measurements for both position and orientation.

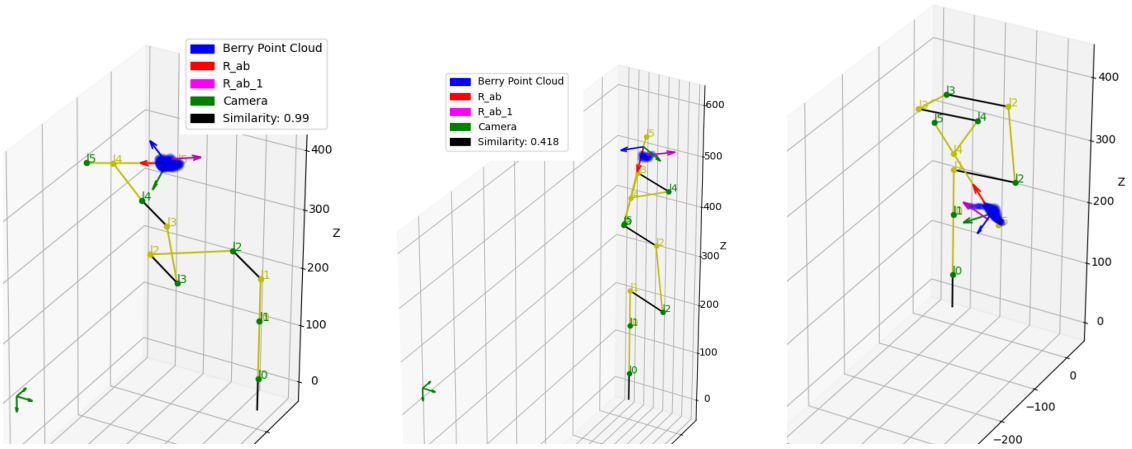
Indeed, through inverse kinematics, and given that the dimensions of the blackberry are known, it is possible to accurately estimate both its position and orientation. Specifically, the rotation matrix describing the orientation of the last joint of the robotic arm with respect to the camera frame was recorded as the blackberry orientation, while the position vector of the tip of the blackberry was recorded as the blackberry picking position.

Pose estimation accuracy was then computed as the absolute cosine similarity between the predicted unit axis $\hat{\mathbf{a}}$ and the ground truth unit axis $\hat{\mathbf{b}}$, according to:

$$PS = \left| \hat{\mathbf{a}} \cdot \hat{\mathbf{b}} \right|$$

where $(\cdot)$ denotes the dot product.

As shown in Fig. 6.8, the method achieved a median pose similarity of 0.9912, with 67% of the tests scoring above 0.9. These values correspond to orientation errors ranging between 8° and 26°.



(a) (b) (c)

Figure 6.9: The figure shows the orientation and position of the real berry (blue point cloud) and the inferred pose, along with the reconstructed arm links. The magenta arrow represents the ground-truth pose, while the red arrow indicates the inverse of the predicted pose for easier visualization.

Fig. 6.9 presents qualitative examples of the estimated pose on blackberries. In each subfigure, the robotic arm holding the fruit is represented by yellow segments, the blackberry point cloud is shown in blue, and the magenta vector indicates the main orientation of the fruit. The blue and green vectors correspond to the other two orthogonal axes of the rotation matrix. The red vector represents the inverse of the estimated pose, which is plotted in this form to avoid overlap between the predicted and ground-truth vectors.

In Subfig. 6.9a, the estimated pose closely matches the ground truth, demonstrating high accuracy and precision. In contrast, Subfig. 6.9b illustrates a failure case, where the predicted pose is nearly orthogonal to the true pose. This misalignment is consistent with the appearance of the point cloud, which collapses into a nearly circular shape due to occlusion in that instance, leading to a failure of the algorithm. This example corresponds to one of the worst cases shown in Fig. 6.8. Although the median similarity is 0.99, a small number of cases exhibit values close to 0.

Finally, Subfig. 6.9c presents a case in which the overall orientation of the prediction is acceptable, but the direction of the estimated vector is completely inverted.

6.6.4 Blackberry Plucking Point Estimation

Accurately identifying the plucking point is crucial for blackberries, as the fruits must be grasped by the gripper without causing damage. The robotic system therefore

requires both a precise location and orientation. Two approaches were evaluated.

In the first approach, given the point cloud and the estimated pose, the plucking point was inferred by approximating the fruit dimensions. However, tests showed that this method was unsatisfactory. As a result, a second method based on ellipsoid fitting was developed. Considering the morphology of blackberries, the optimal plucking point was estimated by fitting the best ellipsoid to the isolated point cloud. The ellipsoid is defined by the following equation:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{b^2} = 1$$

where the parameter vector is

$$\mathbf{v} = \begin{bmatrix} a \\ b \\ x_0 \\ y_0 \\ z_0 \end{bmatrix}.$$

These parameters intuitively define the ellipsoid's center coordinates in 3D space (x_0, y_0, z_0), while a and b represent the semi-axis lengths along the major and minor axes, respectively. No parameter c is included, as the ellipsoid is assumed to be symmetric about the minor axis.

Figure 6.10 illustrates the fitted ellipsoid (red surface) overlaid on the fruit point cloud (blue dots).

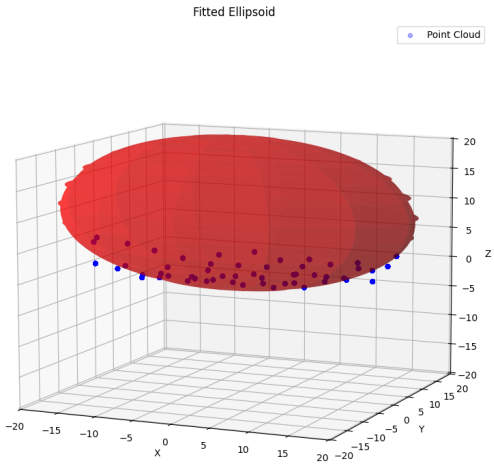


Figure 6.10: Example of ellipsoid fitting to a blackberry point cloud. The blue dots represent the isolated 3D point cloud of the fruit, while the red surface corresponds to the fitted ellipsoid. This fitted ellipsoid is used to estimate the fruit's center and orientation for determining the optimal plucking point.

This method proved particularly effective, consistent with prior studies on food volume estimation [156]. As shown in Fig. 6.11, the use of the fitted ellipsoid's center reduced localization error to a median of 9.29 mm with a standard deviation of 2.52. In contrast, relying solely on the point cloud's centroid resulted in a median error of 19.93 mm and a standard deviation of 11.01 (Fig. 6.11).

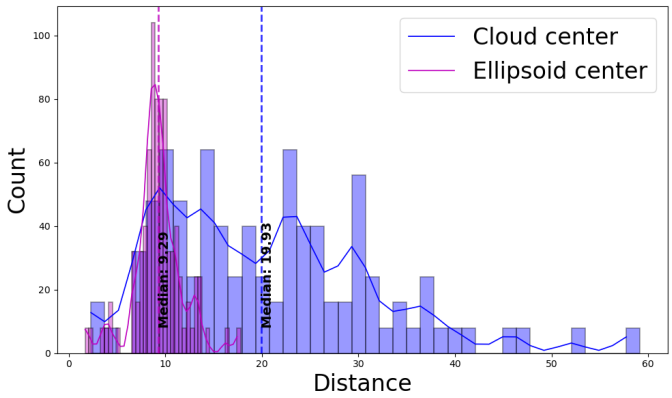


Figure 6.11: Localization error comparison between using the center of the fitted ellipsoid and the center of the point cloud. The dashed line indicates the median error.

Chapter 7

Robotic Manipulation

7.1 Problem Statement

The tested robotic arm was chosen primarily because of its affordability, as cost remains one of the major barriers to the adoption of robotic automation in agriculture [113]. Commercial robotic platforms often come with extensive software support, including integration with the Robot Operating System (ROS) and detailed Unified Robot Description Format (URDF) models, which facilitate simulation, control, and development. In contrast, low-budget or custom-built robotic arms frequently lack such resources, making them less accessible for research and deployment. This absence of standardized tools and documentation creates significant challenges for testing and validating robotic solutions in realistic agricultural environments. As a result, alternative methods must be investigated to overcome these limitations and to explore how low-cost platforms can contribute to broader and more sustainable adoption of robotics in farming. The overall setup was thoroughly tested both in synthetic environments and in real-world scenarios.

7.2 Original Contributions and Methodological Adaptations

This chapter builds upon well-established methodologies for robotic manipulation and pick-and-place operations, which have been extensively investigated in industrial automation contexts. The original contribution of this work does not reside in the development of novel kinematic or motion planning algorithms per se, but rather in their adaptation, integration, and deployment within a lightweight manip-

ulation pipeline designed to operate on a low-cost robotic platform for autonomous agricultural harvesting.

Specifically, classical formulations for forward and inverse kinematics, including the Product of Exponentials representation and iterative Newton-Raphson optimisation, were adopted as baseline approaches and subsequently tailored to address the limitations imposed by the robotic arm's proprietary API, which supports joint position control only and does not provide feedback signals. In this context, a custom inverse kinematics strategy was implemented by incorporating constrained joint limits and task-space-informed initialisation through a database of precomputed configurations in order to improve solver convergence within an open-loop architecture.

In addition, a simplified self-collision avoidance mechanism based on cylindrical joint modelling and a structured robot description file were introduced to provide a lightweight alternative to standard URDF representations, thereby enabling reliable trajectory generation and execution on hardware platforms that would otherwise be unsuitable for precise manipulation tasks in agricultural environments.

7.3 Controller pipeline

The ROS framework provides a variety of kinematics solvers, planners, and related tools. Building on this premise, we developed an ad hoc kinematic module using the available Application Programming Interface (API) for our robotic arm, which only supported joint position control. This limitation arises because the API allows specifying the angle of each joint but not the speed of rotation. Furthermore, due to the same API restrictions, the system operates in an open-loop configuration, since no feedback can be incorporated.

The inverse kinematics is formulated using the Product of Exponentials (PoE) representation and solved iteratively with a Newton–Raphson (NR) method. We adapted existing libraries from [157] to include joint constraints, preventing both self-collisions and external collisions, and to improve NR convergence through a preliminary task-space mapping. For trajectory planning, polynomial fitting of different degrees is employed depending on the task: a cubic (third-degree) polynomial is used for the berry-approach phase, while a linear (first-degree) polynomial is sufficient for the ingestion phase

The overall controller pipeline is illustrated in Fig. 7.1. Once the vision, pose, and keypoint estimation module produces a reachable output, the target object's transformation matrix is passed to the controller. Based on this matrix, the con-

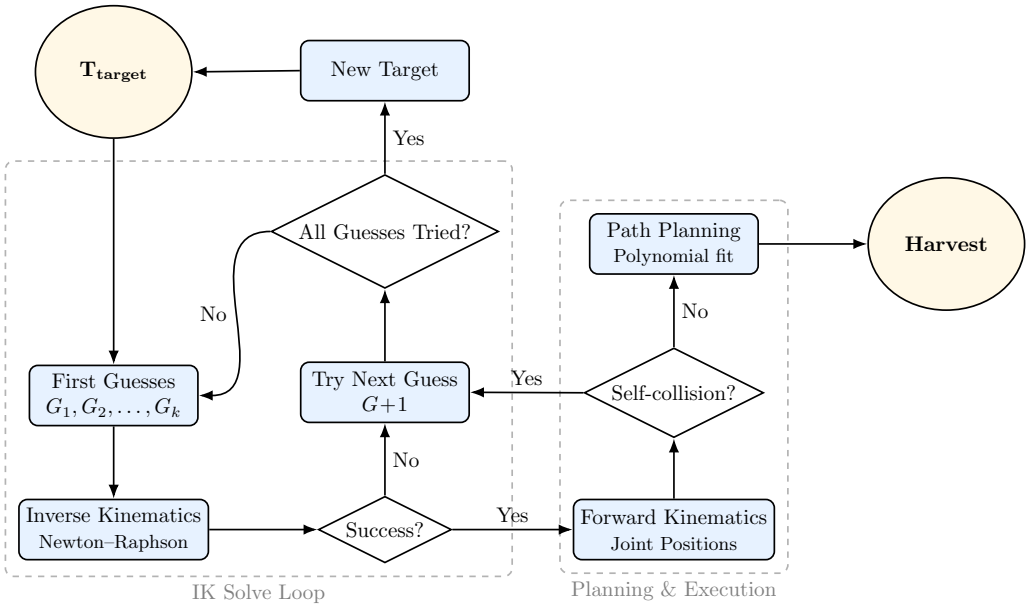


Figure 7.1: Flowchart of the manipulation pipeline. Starting from the target transformation matrix T_{target}, the controller generates initial guesses for the inverse kinematics solver. If a valid solution is found, forward kinematics are computed and joint positions are checked for self-collision. A successful configuration proceeds to path planning and execution (harvest), while failures lead to trying alternative guesses or assigning a new target.

troller prepares a set of k initial guesses for the inverse kinematics solver. If no solution is found, the algorithm iterates through different guesses until either a valid solution is obtained or all guesses are exhausted, in which case a new target is assigned.

When inverse kinematics succeeds, forward kinematics is computed to determine each joint's position, which is then checked for potential self-collisions. If a collision is detected, the pipeline resumes with a different initial guess. Otherwise, the path planning module computes an optimal trajectory to the target using polynomial fitting. The same procedure is applied recursively to all intermediate points along the planned path.

The overall picking process is divided into five steps. The first step, *Aproach*, positions the arm close to the blackberry and aligns it with the fruit's orientation. Next, during *Ingestion*, the arm moves to guide the blackberry into the gripper with steady, smooth, and precise motion. Once the fruit has been secured, *Inflation* ensures that the gripper holds the blackberry firmly. This is followed by *Detach*, where the arm pulls the blackberry from the branch. Finally, in *Discard*, the arm moves to a predefined location to release the blackberry into a dedicated punnet.

7.3.1 Arm Modeling

The kinematic model is governed by the PoE formulation for inverse kinematics. In contrast, the Denavit–Hartenberg (DH) representation is used only to assign reference frames to each link for visualization and modeling. Because of this separation, the robot and its properties can be described in a structured YAML file.

This file contains both general communication parameters (e.g., baud rate) and robot-specific details, such as joint limits, link lengths and dimensions, axis centers and orientations, and link diameters. In essence, it provides a complete description of the robotic arm and its kinematic chain.

The approach is conceptually similar to using URDF files in ROS packages but offers an alternative way to represent the same information. Creating such a file is relatively simple: for commercially available low-cost robotic arms, manufacturers may already provide it, while for custom-built robots, the necessary measurements can easily be taken and recorded

7.3.2 Kinematics: Forward and Inverse

Following the principle of creating a minimal architecture and building on the formulation described by Lynch and Park [157], kinematics are represented using the PoE approach. This method expresses the position and orientation of the end-effector relative to the base as a product of matrix exponentials, each corresponding to a screw motion. Unlike the DH convention, PoE does not require intermediate link frames; only the base frame and the end-effector frame need to be defined. Specifically, for an n-DOF open-chain robotic arm, the forward kinematics can be written as in Equation (7.1):

$$T(\theta) = e^{[S_1]\theta_1}, e^{[S_2]\theta_2}, \cdots, e^{[S_n]\theta_n}, M \tag{7.1}$$

where:

- $T(\theta) \in SE(3)$ is the forward kinematics (end-effector pose as a homogeneous transformation),
- $M \in SE(3)$ is the home configuration of the end-effector (pose when $\theta = \mathbf{0}$),
- $[S_i] \in \mathfrak{se}(3)$ is the matrix representation of the twist associated with the i-th screw axis,
- $\theta_i \in \mathbb{R}$ is the joint variable (e.g., the rotation angle for a revolute joint).

Alternatively, the kinematic chain can be expressed as the product of homogeneous transformation matrices associated with each joint, similar to the DH representation, as shown in Equation (7.2):

$$T_{0,n}(\theta);=;T_{0,1}(\theta_1);T_{1,2}(\theta_2);\cdots;T_{n-1,n}(\theta_n), \tag{7.2}$$

Each term $T_{i-1,i}(\theta_i)$ represents the pose of link i relative to link $i-1$, parameterized by the joint variable θ_i. The overall transformation $T_{0,n}(\theta)$ thus maps coordinates from the base frame to the end-effector frame by chaining all intermediate joint transformations.

PoE was chosen over the DH convention because it offers several practical advantages. First, it avoids the strict frame-assignment rules required by DH, since kinematics are defined directly from the screw axes and the end-effector's home configuration. This reduces ambiguity and simplifies modeling. Second, PoE is more geometrically intuitive, as each joint is described directly by its physical motion axis. Third, it provides a unified representation for revolute, prismatic, and even helical joints, while DH treats them separately. Finally, PoE scales more naturally to complex robotic systems with non-standard joints or branching kinematic chains.

Algorithm 1 Newton–Raphson Inverse Kinematics

Require: Target pose T_d, initial guess θ^0, max iterations K, tolerances $\varepsilon_p, \varepsilon_o$

$\theta \leftarrow \theta^0$
for $k = 1$ to K **do**
  $T \leftarrow FK(\theta)$
  $\Xi_b \leftarrow \log(T^{-1}T_d)$
  if $|\Xi_b^{lin}| < \varepsilon_p$ **and** $|\Xi_b^{ang}| < \varepsilon_o$ **then**
    return θ
  end if
  $J_b \leftarrow \text{BodyJacobian}(\theta)$
  $\Delta\theta \leftarrow J_b^{\dagger}, \Xi_b$
  $\theta \leftarrow \theta + \Delta\theta$
end for
return θ

To reach a desired end-effector pose, an inverse kinematics algorithm is required. The Newton–Raphson iterative method was chosen for its simplicity. Algorithm 1 outlines this procedure: starting from an initial guess, the forward kinematics is computed, the error is expressed as a body twist via the matrix logarithm, and an update step is applied using the pseudoinverse (or damped least squares) of the body

Jacobian. This process repeats until the error falls below the given tolerances or the iteration limit is reached.

However, being iterative and joint-space based, the method is sensitive to the initial guess. Two solutions that are close in Cartesian space may correspond to very different points in joint space, making convergence harder. To mitigate this, a database of 6000 precomputed joint configurations was generated. For any given target pose, the 10 closest candidates in Cartesian space are selected as initial guesses. As described in Fig. 7.1, the algorithm attempts these candidates in sequence; if the first guess fails to converge, the process is repeated with the next, up to the tenth. If none succeed, the target point is discarded and the system proceeds to the next command.

7.3.3 Self Collision avoidance

As straightforward as it may seem, once the inverse kinematics are solved, the position and orientation of each joint in space can be easily derived through forward kinematics. Self-collision is then checked by modeling each joint as a cylinder and verifying whether these cylinders overlap. Avoiding self-collision is essential because the implemented kinematics algorithms do not account for the fact that arm joints have physical volume. Other types of collision avoidance are unnecessary in this scenario, since there is no interaction with other humans or robots, and the crop is positioned directly in front of the robotic arm. At worst, the arm might touch or move some leaves or fruits, but the kinematics ensure slow and gentle motion so that the crop is not damaged. In fact, occasionally moving the leaves can even be beneficial, as it may uncover fruits that were previously hidden.

Self-collision avoidance is not required at every stage of the process and is therefore only implemented during the Approach phase. During Ingestion, collisions are impossible by design, while the final point is simply checked for reachability to ensure it lies within the arm's workspace. Similarly, Detach merely reverses the previous motion, and Discard relies on the arm's API, which already provides a collision-free path. The discard position is also predefined to be both reachable and collision-free. These properties significantly reduce computational demands, enabling the algorithm to operate efficiently even on low-power devices.

7.3.4 Path Planning

For self-collision avoidance, path planning is required only during the Approach phase of the picking loop. Since no other objects are present within the robot arm's

workspace, path planning is used solely to ensure a smooth and efficient trajectory, particularly when approaching the target fruit. Given the target point, an additional point offset by the approximate length of the blackberry, and the starting position, a 3D cubic interpolation is applied. Figure 7.2 illustrates this process: the blue markers represent the control points used to fit the curve, while the magenta markers correspond to the eight sampled checkpoints along the trajectory. For each checkpoint, the complete kinematics pipeline (Fig. 7.1) is executed to guarantee a smooth and collision-free path. The number of checkpoints was carefully tuned to balance trajectory smoothness with computational efficiency.

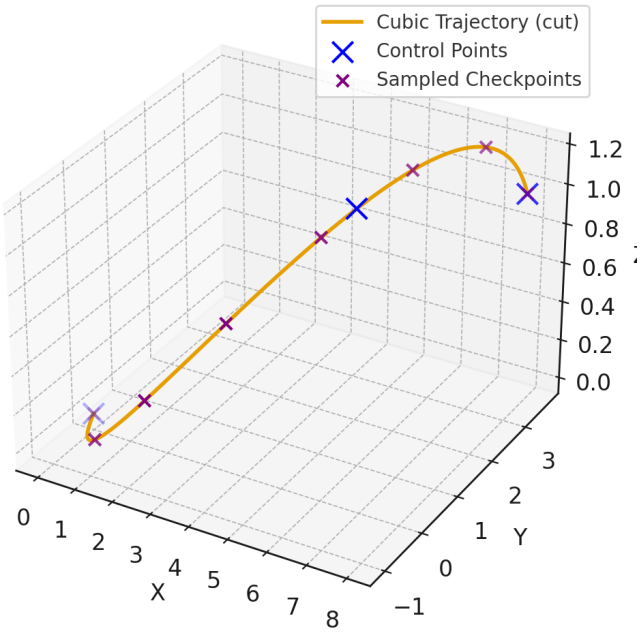


Figure 7.2: Cubic polynomial trajectory fitted through selected control points. The blue markers represent the chosen control points (p_0, p_1, and p_3), while the purple markers indicate the eight sampled checkpoints obtained for $k = 6$. The trajectory is shown only between the two external control points, providing a smooth, collision-free path approximation for the robot arm.

7.4 Experimental Results

The manipulation module was extensively tested to evaluate both the robustness of the control system and the precision of the manipulator. For the arm, the tests revealed positioning errors within 1 mm, consistent with the datasheet specifications, representing competitive performance given the hardware's cost (approximately £1,000). However, in 7 out of 10 sets of 25 repetitions, the arm sporadically

deviated to a random location (indicated by black arrows in Fig. 7.3), with offsets ranging between 1 and 4 mm from the target position.

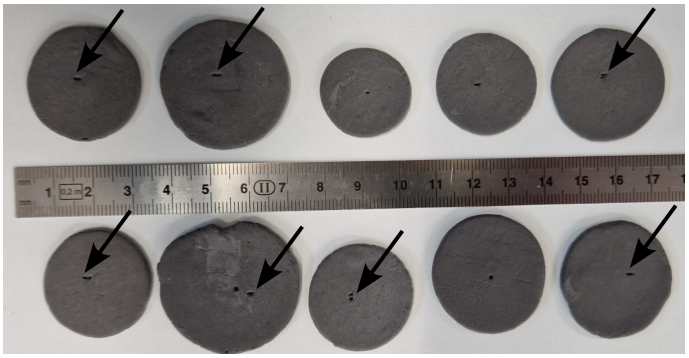


Figure 7.3: Marks on clay disks for accuracy tests of the arm.

The control system was tested throughout development and later assessed in a more systematic and objective manner as part of the task benchmarking pipeline. The overall benchmarking methodology will be described in the following chapter. These evaluations clearly show that the hardware is insufficient for certain tasks, particularly ingestion, where steady and smooth motion is required. In this case, the arm exhibited jittery movements that prevented the task from being completed successfully.

Chapter 8

Experimental Protocols

8.1 Problem Statement

A unified and robust procedure for evaluating robotic harvesting is still lacking in the literature. Existing studies often rely on ad-hoc evaluation criteria, task-specific success rates, or qualitative descriptions, which makes systematic comparison across different approaches challenging. This absence makes it difficult not only to compare different studies but also to identify strengths and weaknesses in an objective manner, even for the researchers themselves. To address this issue, the evaluation framework proposed for benchmarking robotic competitions [158] is adopted, in which experiments are classified via Task-based Benchmarking (TB). TB enables structured, repeatable, and implementation-agnostic assessment of performance at the level of individual sub-tasks in the harvesting pipeline.

8.2 Original Contributions and Methodological Adaptations

This chapter builds upon established task-based benchmarking methodologies commonly employed in robotic competitions and manipulation studies. The primary contribution of this work lies in their adaptation and formalisation within the context of autonomous agricultural harvesting.

In particular, a structured task-based experimental protocol was defined to enable the systematic evaluation of each sub-task of the harvesting pipeline—including perception, pose estimation, manipulation, and grasping—as well as of the overall picking process. This framework allows subsystem-level performance to be anal-

ysed independently while also capturing the impact of error propagation across the integrated pipeline.

The proposed protocol therefore provides a reproducible and implementation-agnostic evaluation procedure that may support future benchmarking and ablation studies in robotic harvesting, facilitating objective comparison across different system architectures and contributing towards more standardised assessment practices in agricultural robotics.

8.3 Task-based Experiments

TB provides a structured methodology for assessing robotic performance by decomposing complex operations into measurable sub-tasks. Each sub-task is evaluated according to its degree of successful completion (e.g., complete, partial, unsatisfactory), independent of the particular subsystems responsible for execution [159]. This abstraction makes the evaluation process implementation-agnostic, allowing systems with different architectures to be compared on a common ground. In particular, a subtask is considered *Complete* when it is successfully executed as expected. It is classified as *Partial* when the task is not completed optimally but still does not compromise subsequent tasks. For example, this occurs when the gripper partially ingests the berry or inadvertently ingests a leaf along with the berry. While these cases are suboptimal, they do not prevent the berry from being effectively detached from the plant. The final category, *Unsatisfactory*, applies when the task is not completed and it is impossible to proceed with the pipeline.

A primary strength of TB lies in its modularity: by isolating individual stages of the robotic pipeline researchers can identify where failures occur and how they influence downstream performance. For instance, the Fields2Benchmark framework for agricultural robotics demonstrates how coverage path planning can be decomposed into field decomposition, route planning, and path planning, enabling detailed analysis of performance at each level [160]. Furthermore, TB supports cross-system comparability and reproducibility, as shown by the YCB object se [161].

Despite its utility, TB also presents some limitations. First, since it focuses on isolated sub-tasks, this may obscure emergent behaviors that arise from subsystem interactions, such as cascading errors [162]. Second, the value of TB depends heavily on well-specified task protocols; inconsistencies in task definitions may hinder reproducibility or in manipulation benchmarks [161]. Finally, TB faces a trade-off: while it enables fine-grained evaluation of specific subtasks, success in these isolated contexts does not always generalize to the complexities of unstructured, real-world

environments.

Despite these limitations, TB has become a cornerstone of benchmarking efforts across multiple domains of robotics. In agricultural robotics, competitions such as ACRE have highlighted the importance of standardized benchmarking frameworks for evaluating weeding and harvesting tasks under realistic conditions [163]. In manipulation, the YCB dataset and REPLAB [164] have promoted reproducibility by providing low-cost, standardized testbeds.

When applied to autonomous blackberry harvesting, TB enables a systematic examination of how sub-tasks are executed, how errors propagate across the pipeline, and how individual components contribute to or constrain overall system performance. For each test, by employing discrete scoring schemes (e.g., 0 = failure, 1 = partial success, 2 = success), TB provides quantitative insights into subsystem performance while maintaining comparability with broader robotic evaluation practices.

8.3.1 Task Decomposition

In task benchmarking, a pivotal step is the definition of sub-tasks. Fig. 8.2 illustrates how this subdivision has been applied to the blackberry harvesting pipeline. The identified sub-tasks are associated with the main modules—vision, arm, and gripper—and closely follow the structure of the control pipeline shown in Fig. 8.1. More generally, this division expands the abstract food-handling flow chart proposed in [165], with the addition of a vision component specific to this application. Although in some cases it is not possible to unambiguously assign a task to a single subsystem, this mapping is often feasible and provides valuable insights into the role and performance of each module within the overall process.

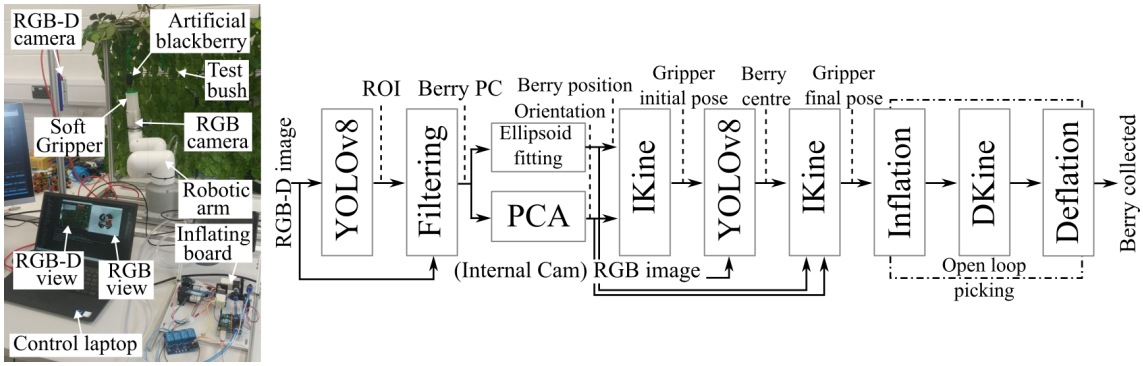


Figure 8.1: The hardware of the blackberry robot (left) and the control pipeline (right).

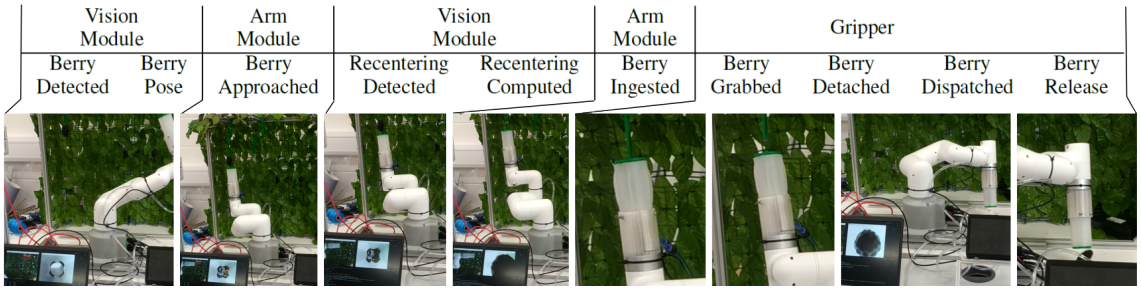


Figure 8.2: The overarching harvesting task is split into several sub-tasks and assigned to individual modules (hardware and software).

8.3.2 Test Benchmarking Setup

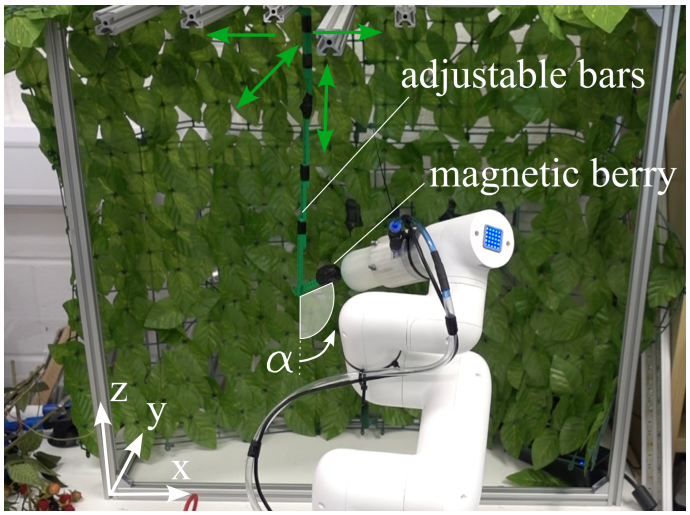
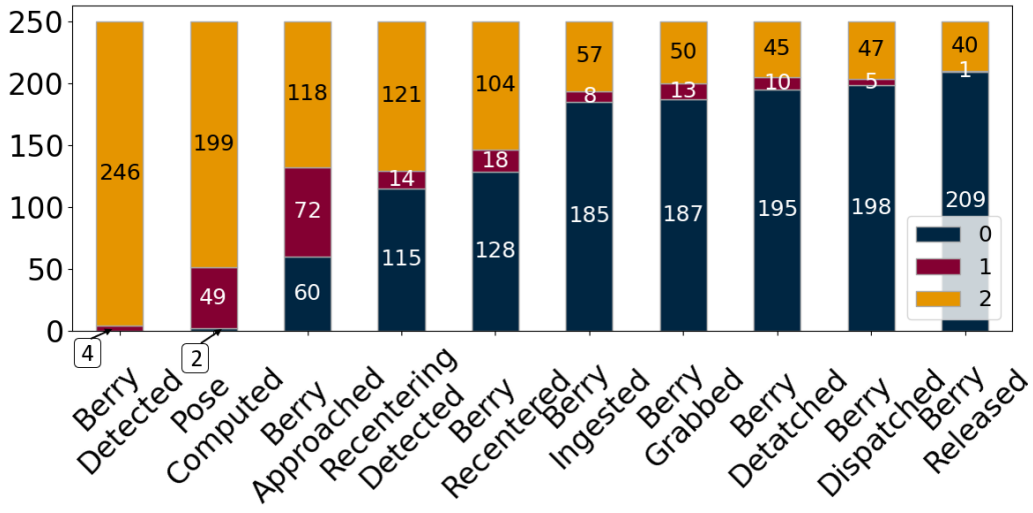


Figure 8.3: The artificial bush used in task benchmarking with the berry and 3d printed stems.

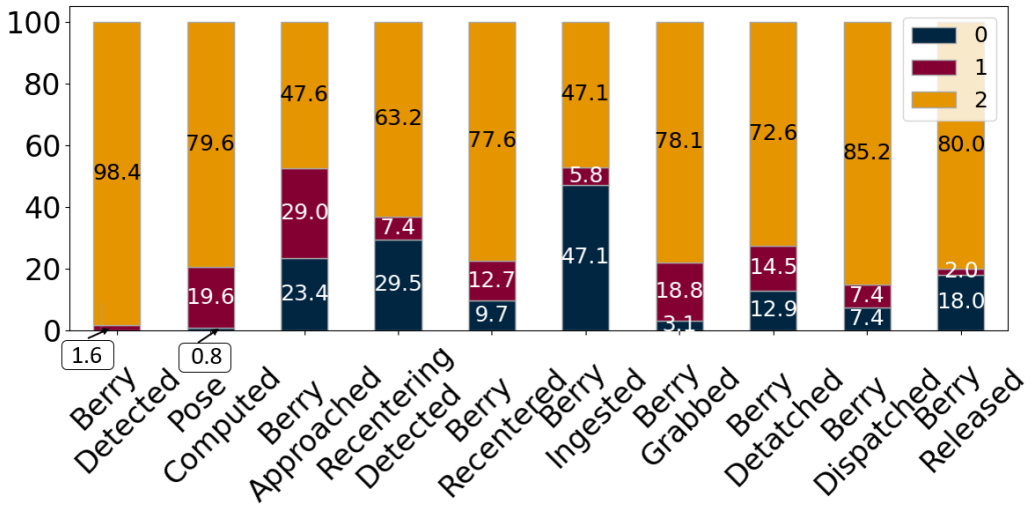
To evaluate the system, an artificial berry was positioned within the robotic arm's workspace using a controlled grid pattern (Fig. 8.3). A total of 250 trials were carried out, spanning four distinct height levels of the bush (560 mm, 440 mm, 320 mm, and 220 mm) and four berry orientations (0, $\pi/4$, $\pi/2$, and $3\pi/4$). This setup ensured a systematic exploration of the arm's performance across varying spatial positions and orientations.

8.4 Performance Results

The performance of the system on sub-tasks is presented in Fig. 8.4, where the top chart reports absolute scores (number of successes, partial successes, or failures



(a)



(b)

Figure 8.4: Comparison between the distribution of overall (a) and relative percentage (b) scored points. In Fig. (b) the percentage for each individual step was calculated using as the sample size the sum of correctly and almost correctly completed trials.

based on the 250 attempts) while the bottom chart reports relative scores (values relative to the successes and partial successes of the previous step). The absolute scores decline as the pipeline progresses, highlighting the relationships between two consecutive sub-tasks. Early failures in the pipelines prevent a proper investigation of the subsequent steps, and these observations suggest employing an experimental approach that includes the user in the loop: by having the chance to correct failures of the pipeline, areas of improvement could be identified with fewer attempts, and the individual components assessed in a more relevant way.

Table 8.1: Task performance metrics for each tested position and orientation. The table reports the relative percentage of successful trials for each configuration. The Overall Success Rate (OSR) column summarizes the total percentage of successful picking operations. The reported 95% confidence intervals correspond to binomial Wilson intervals computed on the overall score.

Spatial Parameters		Vision and Arm results						Gripper metric results				OSR
Z Position (mm)	Orientation α	Det.	Pose	Appr.	Rec. Det.	Rec.	Ing.	Grab.	Det.	Disp.	Rel.	
560	0	100	100	72	95	90	57	83	75	75	67	32
	$\pi/4$	100	76	64	82	75	40	50	75	43	50	8
	$\pi/2$	100	64	12	38	60	0	-	-	-	-	-
440	0	96	100	80	86	95	75	87	75	100	93	52
	$\pi/4$	100	64	29	72	69	33	100	100	75	50	8
	$\pi/2$	100	72	24	22	80	0	-	-	-	-	-
320	0	88	88	68	69	63	66	92	92	100	75	36
	$\pi/4$	100	68	24	21	18	25	100	100	100	100	4
	$\pi/2$	100	64	24	33	100	0	-	-	-	-	-
220	$3\pi/4$	100	100	75	89	100	47	50	50	100	100	17
Overall score		98.4	79.5	47.4	62.9	77.4	46.7	76.6	70.9	85.2	78.5	15.7
95% Confidence Interval		[96.8–99.4]	[74.5–84.0]	[41.3–53.6]	[56.8–68.7]	[72.2–82.0]	[40.6–52.9]	[71.4–81.3]	[65.1–76.2]	[80.4–89.1]	[73.4–83.0]	[11.0–19.8]

The relative scores of Fig. 8.4 highlight bottlenecks of the pipeline.By computing the percentage values using as the sample size the sum of successful and partially successful trials from the preceding task, each score isolates the performance of the current stage, effectively removing the influence of errors propagated from previous steps. The initial detection and pose estimation part showed success ranging from about 80% to 100%, and complete failures lower than 2%. Similar consistency has been found in the gripper result, with success rates ranging from 70% to 85%. The three principal sources of failures have been identified in the approach of the berry, detecting for re-centering, and ingestion sub-tasks. The different camera resolution, point of view, and illumination are all known sources of disturbances for domain generalization of learned approaches [143], but the berry approach and ingestion issues are less investigated. Although we did not perform a formal evaluation, our educated guess is that the arm's movements in task space are less accurate for specific poses of the grasped berry. In other words, although the arm can kinematically solve most of the requested trajectories, motor resolution and practical constraints impaired the proper execution of several picking actions.

This observation found additional evidence when the results are split by position and orientation of the berry (see reference frame in Fig. 8.3) as presented in Table 8.1. Specific poses (e.g. $z = 400$mm and $\alpha = 0$) of the berry showed a constant level of performance of the different subsystems, but when the orientation of the berry was tested in the horizontal plane ($\alpha = \pi/2$rad), we registered an abrupt drop in the approaching and ingestion sub-task scores. It is worth mentioning that such berry poses, and the relative trajectories required for approaching and ingestion, are kinematically feasible but close to the task-space of the arm (due to self-collision con-

straints). Similar issues have been highlighted for positions close to the workspace of the arm, i.e., for $z = 560$mm and $z = 320$mm, and orientation different from $\alpha = 0$. Also in this case, self-collisions and trajectory accuracy in task-space appear to be a significant limit to the performance of the system.

By analyzing the results of the vision system with respect to the position and orientation of the berry (Table 8.1), the detection appeared robust with respect to the position, while the orientation score drops to a lowest of 64%. This issue arose because the vision module struggled when the berry appeared too symmetrical, making it challenging to accurately compute its pose. Although the vision performed to a significant degree of success, ranging from 88% to 100% in detection across different poses, being this task the initial part of every pipeline will require higher and robust degrees of success.

Eventually, the soft gripper demonstrated a success rate across poses ranging from 50% to 100%, with an overall score of 76.6% (Table 8.1). Causes of failure ranged from random hardware failures due to the prototyping nature of the device (e.g. compressor did not initiate the inflation) to loss of grips during arm movements. Partial grasps, i.e. cases in which the gripper was not perfectly positioned around the berry, still resulted in successful grasps, confirming the advantages provided by compliant grippers despite an open-loop gripping approach.

Chapter 9

Conclusions and Future Work

9.1 Contributions and Limitations

This thesis has examined the integration of AI-based perception and low-cost robotic manipulation for the autonomous harvesting of underexplored crops. The primary goal was to assess the feasibility of an AI-driven vision framework for robotic harvesting of high-value crops. A secondary objective was to evaluate the potential of foundational models and state-of-the-art architectures to reduce computational demands and address data scarcity challenges.

The framework developed for edible flowers, FloralAI, demonstrated that the proposed approach can be applied across different species and varieties, highlighting its robustness and adaptability. Its extension to blackberries further confirmed that the pipeline can be transferred with only minimal adjustments. The incorporation of general-purpose and foundation models such as YOLOv5 and SAM enabled a resource-efficient strategy, leveraging public datasets and zero-shot field data. Remarkably, the detection and segmentation of ripe blackberries were achieved exclusively with online, out-of-domain data, thereby validating the results obtained from the edible flower case study.

A central methodological contribution was the development of crop-specific plucking point estimation strategies. For edible flowers, PCA-based pose estimation was combined with diameter inference to predict accurate plucking points. For blackberries, PCA pose estimation was integrated with ellipsoid fitting to localize and orient fruits within clustered canopies. These approaches proved effective under constrained data conditions and demonstrated the adaptability of geometric methods to agricultural perception tasks.

Beyond harvesting, the pipeline, particularly the integration of YOLO and SAM,

shows strong promise as a dataset annotation tool. This method could serve as a ground-truth annotation engine to support the development of future architectures. Since high-quality annotations remain a major bottleneck in training efficient neural networks, the approach described here provides a scalable and reliable solution for generating training data.

This work also presents, for the first time, a fully evaluated autonomous blackberry-picking robot equipped with a soft robotic gripper. The evaluation covered the entire pipeline, combining component-level and task-oriented analyses. Autonomous harvesting trials with berries in diverse orientations and positions demonstrated the feasibility of the approach and the effectiveness of the proposed system. These results also reinforced the findings from edible flower harvesting. Overall, the study systematically analyzed and documented the complete picking process, highlighting both its strengths and limitations. Furthermore, it provides the first comprehensive evaluation of a blackberry-harvesting robot based on generic, replicable, and adaptable testing procedures.

The evaluation also revealed several broader insights. While general-purpose robotic platforms are useful for testing, they are not optimal for harvesting. A custom-designed robotic arm, specifically engineered for agricultural tasks, would provide a more effective solution. At the same time, the implemented control pipeline proved well-suited to the task, showing that complex, computationally expensive algorithms are not always necessary—classical control and kinematic principles can suffice.

With respect to the end effector, the advantages of a soft gripper extended beyond its ability to avoid bruising the crop. Its compliance also helped accommodate minor inaccuracies in perception and control. The evaluation highlighted how crop-specific end effector design is critical to reducing task complexity. For example, the cylindrical design of the tested gripper simplified alignment, since only one axis needed to be aligned with the fruit, eliminating the need for complex gripping strategies. Conversely, in the case of edible flowers, a specialized gripper capable of both holding and cutting the crop would be essential.

Finally, this thesis introduced task-based benchmarking protocols for harvesting evaluation. By standardizing both component-specific and task-oriented assessments, these protocols provide an implementation-agnostic framework for reproducible comparison and future research in agricultural robotics.

Pose estimation assessment proved to be non-trivial. As demonstrated in the case of edible flowers, a functional robotic picking system is essential for fully evaluating pose estimation accuracy, since it provides a reliable ground truth for per-

formance validation. This requirement was addressed in the blackberry harvesting experiments by employing a robotic arm to emulate the crop. In this setup, the kinematics of the arm itself provided precise ground-truth position and orientation data.

However, this approach has inherent limitations. Because it was performed in a synthetic scenario, even when physical twins are designed to closely replicate real conditions, small differences, gaps, and biases relative to the real-world domain inevitably remain. Moreover, critical field factors such as occlusion, environmental variability, and seasonal fluctuations can significantly influence detection accuracy and overall system performance. Indeed, preliminary in-field tests with the blackberry harvester confirmed that variations in lighting conditions had a substantial impact on the performance of the vision module.

9.2 Comparative Analysis of Target Crops

The two crop types investigated in this thesis, blackberries and edible flowers, illustrate how biological and structural characteristics strongly influence the automation potential of agricultural tasks. Blackberries proved relatively easier to automate due to their natural detachment properties and the structural uniformity of berry clusters. Once ripe, the fruit can be removed with minimal force, reducing the likelihood of damage and simplifying both gripper design and control strategies. Moreover, the clustered arrangement of berries provides repeated opportunities for successful harvesting actions within a single reach.

By contrast, edible flowers presented a far more demanding challenge. Their fragile petals, slender stems, and high variability in shape and orientation require not only precise perception but also highly accurate end-effector positioning and cutting. Small deviations can result in damaged flowers, rendering them unsuitable for commercial use. This comparison highlights a fundamental principle: while some crops offer natural affordances that align with robotic harvesting, others require specialized solutions that push the limits of current perception, manipulation, and control technologies.

9.3 Model Transferability

The experimental results further emphasize the partial transferability of AI-based perception models in agriculture. Detection models trained on one crop demon-

strated moderate generalization when applied to another, suggesting that deep-learning-based detectors are able to capture shared visual cues such as color, texture, and basic geometry. This level of transfer is particularly encouraging as it reduces the cost and time required to train models from scratch for each new crop.

However, pose estimation and manipulation strategies exhibited far lower levels of transferability. Differences in geometry, surface texture, orientation, and occlusion patterns require substantial crop-specific tuning. For example, the strategies used to localize and approach a blackberry cluster could not be directly applied to edible flowers, where millimeter-scale accuracy is required. These results point to an important trade-off: while perception modules may benefit from cross-crop generalization, reliable harvesting performance ultimately demands tailored manipulation strategies.

9.4 Generalizability of the Proposed Pipeline

The proposed robotic pipeline was deliberately designed with modularity as a central feature, enabling partial reuse of perception, control, and manipulation components across different crops. This modular approach proved advantageous in practice, as it reduced the engineering overhead when transitioning from edible flowers to blackberries. Components such as object detection and segmentation were reused with minimal adaptation, providing a strong foundation for flexible system design.

Nevertheless, the results also underline the limitations of full generalization. Agricultural environments are highly variable, with differences in canopy density, lighting conditions, and fruit or flower morphology. These factors limit the extent to which a single pipeline can be universally applied. The evidence suggests that future robotic harvesting solutions should embrace a hybrid philosophy: leveraging modular and reusable components where possible, while simultaneously incorporating crop-specific adaptations where necessary. Such an approach would balance scalability with robustness, enabling the gradual expansion of robotic harvesting systems to a broader range of crops.

In addition, the studies and tests presented in this thesis make it clear that the success of robotic harvesting does not depend solely on advances in robotics and AI. Progress must also come from the agricultural side, through dedicated crop research. Factors such as plant architecture, growth patterns, and cultivation methods play a critical role in determining the feasibility of automation. For example, breeding or training crops with more uniform structures, accessible fruit placement, or reduced canopy density could substantially lower the complexity of robotic harvesting tasks.

Thus, a true pathway toward successful agricultural automation requires a joint effort between robotics research and agricultural science.

9.5 Future Perspectives

9.5.1 Scalable Dataset Collection

One of the clearest bottlenecks identified in this thesis is the limited availability of diverse and representative datasets. Current models often fail to generalize because training data does not adequately capture the variability of real-world conditions. Future research should therefore prioritize the development of large-scale, open-access agricultural datasets spanning multiple crops, environments, and growth stages.

Automating the annotation process will be key to scaling up dataset creation. The integration of advanced segmentation tools such as SAM could drastically reduce the time and effort required for annotation, while also ensuring greater consistency and accuracy. Such tools could serve both as accelerators for model training and as benchmarks for validating emerging algorithms. By addressing the data bottleneck, the research community can enable broader generalization and accelerate the pace of innovation in agricultural robotics.

9.5.2 Robot-Assisted Dataset Generation for Future Benchmarking

As discussed in the previous chapters, one of the main limitations encountered in this work concerns the absence of standardised datasets for benchmarking perception and pose estimation algorithms in robotic harvesting applications. This limitation significantly restricts the possibility of performing fair comparative evaluations between different detection and manipulation pipelines. The issue largely stems from the industrial nature of many robotic harvesting solutions, where datasets are often proprietary and therefore not publicly available. Furthermore, acquiring reliable ground-truth annotations—particularly with respect to the three-dimensional position and orientation of target fruits—remains a significant challenge.

To address this limitation, an ongoing research effort has been initiated towards the development of a dedicated dataset for blackberry detection, keypoint extraction, and pose estimation. In current robotic picking applications, ground-truth pose information is typically obtained through manual annotation. However, due to

$$R_1 = \begin{bmatrix} -0.3824 & -0.9245 & -0.0131 \\ -0.4132 & 0.1841 & -0.892 \\ 0.8272 & -0.3353 & -0.4525 \end{bmatrix} \quad R_2 = \begin{bmatrix} 0.5741 & -0.1478 & -0.8053 \\ -0.7142 & -0.5714 & -0.4043 \\ -0.4004 & 0.8073 & -0.4336 \end{bmatrix} \quad R_3 = \begin{bmatrix} -0.1844 & -0.7730 & 0.6071 \\ -0.5002 & 0.6055 & 0.6190 \\ -0.8460 & -0.1895 & -0.4983 \end{bmatrix}$$

$$t_1 = \begin{bmatrix} 0.3714 \\ -0.2856 \\ 1.2882 \end{bmatrix} \quad t_2 = \begin{bmatrix} -0.3196 \\ -0.2698 \\ 1.2339 \end{bmatrix} \quad t_3 = \begin{bmatrix} 0.1251 \\ 0.0916 \\ 0.8196 \end{bmatrix}$$

Figure 9.1: Examples of the data gathered in the dataset. The first row represents the images of the blackberries, while the second and third rows show, respectively, the rotation matrices and translation vectors for the tip of each blackberry. These data correspond to the ground-truth picking point and pose of the blackberries.

the inherently three-dimensional nature of this task, such procedures are both time-consuming and prone to inaccuracies, especially when precise pose estimation is required. This lack of accurate and standardised data ultimately limits the possibility of performing systematic benchmarking and comparative studies across different robotic harvesting approaches.

Building upon the methodology introduced in Section 6.6.3, the proposed approach aims to generate annotated RGB images and corresponding point clouds in which ground-truth information is obtained autonomously. A synthetic replica of a blackberry is mounted on the end-effector of a robotic manipulator and randomly moved throughout the robot workspace. By leveraging the inverse kinematics of the manipulator, it becomes possible to accurately record the ground-truth position and orientation of the blackberry in each acquired frame without manual intervention.

The dataset currently includes approximately 2000 annotated instances and is intended to support the future training and evaluation of detection, keypoint extraction, and pose estimation algorithms. As illustrated in Fig. 9.1, artificial foliage has been introduced in the background to increase scene complexity and simulate partial occlusions, which represent one of the main challenges in in-field detection tasks. This setup allows for the controlled reproduction of realistic harvesting conditions while preserving accurate ground-truth pose and picking-point information.

(a) (b)

Figure 9.2: Example of the automatically generated ground-truth annotations. The red bounding box represents the projected 2D bounding box of the blackberry, the yellow points indicate the keypoints, and the red, green, and blue segments depict the pose vector describing the orientation of the fruit with respect to the camera reference frame.

Preliminary visual examples of the results are shown in Fig. 9.2. In particular, the red bounding box associated with each blackberry is obtained by projecting the corresponding three-dimensional bounding box onto the two-dimensional image plane. Since the physical dimensions of the blackberry, along with the orientation and position of its tip, are known, it is possible to obtain both 3D and 2D bounding box annotations. This is expected to be beneficial not only for standard 2D detection tasks based on computer vision techniques, but also for 3D detection within the point cloud space. Moreover, the yellow dots represent the keypoints assigned to the blackberry, while the red, green, and blue segments indicate the pose vector describing the orientation of the fruit with respect to the camera reference frame.

Once completed, the proposed dataset is expected to facilitate the development and systematic evaluation of perception modules for robotic harvesting applications. More importantly, it may provide a standardised resource for benchmarking and comparative analysis of different detection and pose estimation pipelines. In this way, it is intended to contribute to improving consistency and reproducibility across future robotic picking systems, and may be further extended to include additional crop types.

9.5.3 Multi-Modal Perception

A second promising direction lies in the integration of multi-modal sensing technologies. Relying solely on RGB vision, while effective in controlled conditions, is insufficient for tasks that demand richer context. Hyperspectral imaging could enable accurate ripeness assessment and early disease detection; thermal sensing could provide insights into plant stress; and depth cameras could improve the handling of occlusions and overlapping structures.

The fusion of these modalities with AI-based perception would allow robotic systems to move from simple visual recognition toward a more holistic understanding of crop state and environment. Such advances would not only improve harvesting accuracy but also open pathways to new applications such as selective harvesting, yield estimation, and crop health monitoring.

9.5.4 Field Deployment and Real-World Testing

Perhaps the most critical step forward is the transition from controlled laboratory experiments to field deployment. While controlled settings allow for systematic evaluation and repeatability, they do not capture the full complexity of real-world farming environments. Future research should therefore focus on validating robotic pipelines under actual agricultural conditions, where uncontrolled lighting, weather variability, soil unevenness, and frequent occlusions pose major challenges.

Three priorities stand out for successful deployment: first, the development of robust perception pipelines that can withstand varying illumination and partial occlusions; second, the design of efficient models capable of real-time inference and actuation without excessive computational overhead; and third, the demonstration of long-term reliability, ensuring that robotic systems can operate consistently over extended periods. Achieving these goals will be essential to bridging the gap between proof-of-concept prototypes and commercially viable robotic harvesters.

List of published papers

Contributions List of the contribution in the thesis

1 **Artificial Intelligence Vision Methods for Robotic Harvesting of Edible Flowers**, *Fabio Taddei Dalla Torre, Farid Melgani, Ilaria Pertot, and Cesare Furlanello.* In *Plants*, 2024

2 **Toward autonomous blackberry harvesting with a soft gripper and vision-controlled robotic arm**, *Fabio Taddei Dalla Torre, Omar Faris, Philip H Johnson, and Marcello Calisti.* In *2025 IEEE 8th International Conference on Soft Robotics (RoboSoft)*, 2025

Bibliography

[1] Mohamed Ali Mekouar. "15. Food and agriculture organization of the united nations (FAO)". In: *Yearbook of International Environmental Law* 29 (2018), pp. 448–468.

[2] Paolo Malanima and Vera Zamagni. "150 years of the Italian economy, 1861–2010". In: *Journal of Modern Italian Studies* 15.1 (2010), pp. 1–20. doi: 10.1080/13545710903465507. eprint: https://doi.org/10.1080/13545710903465507. url: https://doi.org/10.1080/13545710903465507.

[3] Marina Sorrentino. "11Nutrition". In: *Measuring Wellbeing: A History of Italian Living Standards*. Oxford University Press, Mar. 2017. isbn: 9780199944590. doi: 10.1093/acprof:oso/9780199944590.003.0002. eprint: https://academic.oup.com/book/0/chapter/310366807/chapter-ag-pdf/44497769/book_35949_section_310366807.ag.pdf. url: https://doi.org/10.1093/acprof:oso/9780199944590.003.0002.

[4] Kossi Dodzi Bissadu, Salleh Sonko, and Gahangir Hossain. "Society 5.0 enabled agriculture: Drivers, enabling technologies, architectures, opportunities, and challenges". In: *Information Processing in Agriculture* 12.1 (2025), pp. 112–124.

[5] Jin Yuan, Wei Ji, and Qingchun Feng. *Robots and autonomous machines for sustainable agriculture production*. 2023.

[6] Abderrachid Hamrani et al. "AI and Robotics in Agriculture: A Systematic and Quantitative Review of Research Trends (2015–2025)". In: *Preprints* (Sept. 2025). doi: 10.20944/preprints202509.0408.v1. url: https://doi.org/10.20944/preprints202509.0408.v1.

[7] World Intellectual Property Organization. *Autonomous Devices in Precision Agriculture — Global Overview*. Accessed: 2025-09-10. 2024. url: https://www.wipo.int/web-publications/patent-landscape-report-agrifood/en/7-autonomous-devices-in-precision-agriculture.html#h2-global-overview.

[8] Junior R. Davis. “How Can the Poor Benefit from the Growing Markets for High Value Agricultural Products?” en. In: *SSRN Electronic Journal* (2006). issn: 1556-5068. doi: 10.2139/ssrn.944027. url: http://www.ssrn.com/abstract=944027 (visited on 05/17/2023).

[9] Teng Wang, Huilin Liu, and Zhaohua Wang. “Decomposing the Impact of Agricultural Mechanization on Agricultural Output Growth: A Case Study Based on China’s Winter Wheat”. In: *Sustainability* 17.5 (2025), p. 1777.

[10] Mercedes Ames et al. “107th Annual Meeting of The Potato Association of America, Abstracts and Posters, Prince Edward Island, Canada July 23-27, 2023”. In: *American Journal of Potato Research* 101 (2024), pp. 163–201.

[11] Wanglin Ma et al. “Adoption and intensity of agricultural mechanization and their impact on non-farm employment of rural women”. In: *World development* 173 (2024), p. 106434.

[12] Tamrat Gebiso et al. “Impact of farm mechanization on crop productivity and economic efficiency in central and southern Oromia, Ethiopia”. In: *Frontiers in Sustainable Food Systems* 8 (2024), p. 1414912.

[13] Mobin Amoozad-Khalili et al. “Economic modeling of mechanized and semi-mechanized rainfed wheat production systems using multiple linear regression model”. In: *Information Processing in Agriculture* 7.1 (2020), pp. 30–40.

[14] Xiaojing Ren et al. “Progress in Mechanized Harvesting Technologies and Equipment for Minor Cereals: A Review”. In: *Agriculture* 15.15 (2025), p. 1576.

[15] Cornelis Wouter Bac et al. “Harvesting Robots for High-Value Crops: State-of-the-Art Review and Challenges Ahead”. In: *Journal of Field Robotics* 31 (July 2014). doi: 10.1002/rob.21525.

[16] J Pargi Sanjay et al. “Comparison between manual harvesting and mechanical harvesting”. In: *J. Sci. Res. Rep* 30 (2024), pp. 917–934.

[17] Christopher Lehnert et al. “Autonomous sweet pepper harvesting for protected cropping systems”. In: *IEEE Robotics and Automation Letters* 2.2 (2017), pp. 872–879.

[18] Leonidas Droukas et al. “A survey of robotic harvesting systems and enabling technologies”. In: *Journal of Intelligent & Robotic Systems* 107.2 (2023), p. 21.

[19] P Srinivas et al. “Mechanized harvesting techniques in horticultural crops: A step towards reduction of cost of cultivation”. en. In: *International Journal of Research in Agronomy* 7.9S (Sept. 2024), pp. 940–945. issn: 2618060X, 26180618. doi: 10.33545/2618060X.2024.v7.i9Sm.1636. url: https://www.agronomyjournals.com/special-issue/2024.v7.i9S.1636 (visited on 08/05/2025).

[20] Kristina Sobekova, Michael R. Thomsen, and Bruce L. Ahrendsen, eds. *Market trends and consumer demand for fresh berries*. eng. 7. 2013. doi: 10.22004/ag.econ.164771.

[21] Thomas Daum. "Mechanization and sustainable agri-food system transformation in the Global South. A review". In: *Agronomy for Sustainable Development* 43.1 (2023), p. 16.

[22] Cristina Mitaritonna and Lionel Ragot. "After Covid-19, will seasonal migrant agricultural workers in Europe be replaced by robots?" en. In: *CEPII Policy Brief* (June 2020). Number: 2020-33 Publisher: CEPII research center. url: https://ideas.repec.org//p/cii/cepipb/2020-33.html (visited on 08/07/2024).

[23] Philip Martin. *Farm Labor Shortages: How Real, What Response?* Giannini Foundation of Agricultural Economics. Accessed: August 8, 2024. 2007. url: https://s.giannini.ucop.edu/uploads/giannini_public/67/33/673330c9-c5a1-4664-ade5-6e9b406b8ef3/v10n5_3.pdf.

[24] Yanbin Hua et al. "Recent advances in intelligent automated fruit harvesting robots". In: *The Open Agriculture Journal* 13.1 (2019).

[25] Christopher Lehnert, Christopher McCool, and Tristan Perez. "Lessons learnt from field trials of a robotic sweet pepper harvester". In: *arXiv preprint arXiv:1706.06203* (2017).

[26] Wei Yin et al. "Fruit detection and pose estimation for grape cluster–harvesting robot using binocular imagery based on deep neural networks". In: *Frontiers in Robotics and AI* 8 (2021), p. 626989.

[27] Jianchao Ci et al. "3D pose estimation of tomato peduncle nodes using deep keypoint detection and point cloud". In: *Biosystems Engineering* 243 (2024), pp. 57–69.

[28] Justin Le Louëdec and Grzegorz Cielniak. "3D shape sensing and deep learning-based segmentation of strawberries". In: *Computers and Electronics in Agriculture* 190 (2021), p. 106374.

[29] Anthony L Gunderman et al. "Tendon-driven soft robotic gripper for berry harvesting". In: *arXiv preprint arXiv:2103.04270* (2021).

[30] Taha Samavati, Mohsen Soryani, and Sina Mansouri. "Sparse 3D Perception for Rose Harvesting Robots: A Two-Stage Approach Bridging Simulation and Real-World Applications". In: *arXiv preprint arXiv:2508.00900* (2025).

[31] Yunchao Tang et al. "Recognition and localization methods for vision-based fruit picking robots: A review". In: *Frontiers in Plant Science* 11 (2020), p. 510.

[32] Yue Pan et al. "Panoptic mapping with fruit completion and pose estimation for horticultural robots". In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2023, pp. 4226–4233.

[33] Alexander Kirillov et al. “Segment anything”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, pp. 4015–4026.

[34] Li Jiang et al. “Overview of Agricultural Machinery Automation Technology for Sustainable Agriculture”. In: *Agronomy* 15.6 (2025), p. 1471.

[35] Ruchik Kashyapkumar Thaker. “Advances in Agricultural Robotics: Present Applications, Challenges, and Future Prospects”. In: *IJLRP-International Journal of Leading Research Publication* 4.12 ().

[36] Robert Finger. “Digital innovations for sustainable and resilient agricultural systems”. In: *European Review of Agricultural Economics* 50.4 (2023), pp. 1277–1309.

[37] Chrysanthi Charatsari et al. “Technological innovation and agrifood systems resilience: The potential and perils of three different strategies”. In: *Frontiers in Sustainable Food Systems* 6 (2022), p. 872706.

[38] Robert E Evenson and Douglas Gollin. “Assessing the impact of the Green Revolution, 1960 to 2000”. In: *science* 300.5620 (2003), pp. 758–762.

[39] Marcelo Rodrigues Barbosa Junior et al. “Advancements in agricultural ground robots for specialty crops: an overview of innovations, challenges, and prospects”. In: *Plants* 13.23 (2024), p. 3372.

[40] Qinghua Yang et al. “A review of core agricultural robot technologies for crop productions”. In: *Computers and Electronics in Agriculture* 206 (2023), p. 107701.

[41] Umair Nawaz et al. “AI in Agriculture: A Survey of Deep Learning Techniques for Crops, Fisheries and Livestock”. In: *arXiv preprint arXiv:2507.22101* (2025).

[42] Andreas Kamilaris and Francesc X Prenafeta-Boldú. “Deep learning in agriculture: A survey”. In: *Computers and electronics in agriculture* 147 (2018), pp. 70–90.

[43] Chen Peng et al. “A strawberry harvest-aiding system with crop-transport collaborative robots: Design, development, and field evaluation”. In: *Journal of Field Robotics* 39.8 (2022), pp. 1231–1257.

[44] Kai Junge, Catarina Pires, and Josie Hughes. “Lab2Field transfer of a robotic raspberry harvester enabled by a soft sensorized physical twin”. In: *Communications Engineering* 2.1 (2023), p. 40.

[45] Chenchen Gu et al. “Research progress on variable-rate spraying technology in orchards”. In: *Applied Engineering in Agriculture* 36.6 (2020), pp. 927–942.

[46] Sjaak Wolfert et al. “Big data in smart farming–a review”. In: *Agricultural systems* 153 (2017), pp. 69–80.

[47] Robin Sharma. “Artificial intelligence in agriculture: a review”. In: *2021 5th international conference on intelligent computing and control systems (ICICCS)*. IEEE. 2021, pp. 937–942.

[48] Priyanga Muruganantham et al. “A systematic literature review on crop yield prediction with deep learning and remote sensing”. In: *Remote Sensing* 14.9 (2022), p. 1990.

[49] Abhasha Joshi et al. “Remote-sensing data and deep-learning techniques in crop mapping and yield prediction: A systematic review”. In: *Remote Sensing* 15.8 (2023), p. 2014.

[50] Saeed Khaki and Lizhi Wang. “Crop yield prediction using deep neural networks”. In: *Frontiers in plant science* 10 (2019), p. 621.

[51] Romiyal George et al. “Past, present and future of deep plant leaf disease recognition: A survey”. In: *Computers and Electronics in Agriculture* 234 (2025), p. 110128.

[52] Shaohua Wang et al. “Advances in deep learning applications for plant disease and pest detection: A review”. In: *Remote Sensing* 17.4 (2025), p. 698.

[53] James Daniel Omaye et al. “Cross-comparative review of Machine learning for plant disease detection: Apple, cassava, cotton and potato plants”. In: *Artificial intelligence in agriculture* 12 (2024), pp. 127–151.

[54] Branislava Lalic et al. “Effectiveness of short-term numerical weather prediction in predicting growing degree days and meteorological conditions for apple scab appearance”. In: *Meteorological Applications* 23.1 (2016), pp. 50–56.

[55] Alexander J Bleasdale and J Duncan Whyatt. “Classifying early apple scab infections in multispectral imagery using convolutional neural networks”. In: *Artificial Intelligence in Agriculture* 15.1 (2025), pp. 39–51.

[56] Ishak Pacal et al. “A systematic review of deep learning techniques for plant diseases”. In: *Artificial Intelligence Review* 57.11 (2024), p. 304.

[57] Vinay Vijayakumar et al. “Smart spraying technologies for precision weed management: A review”. In: *Smart Agricultural Technology* 6 (2023), p. 100337.

[58] Roland Gerhards et al. “Advances in site-specific weed management in agriculture—A review”. In: *Weed Research* 62.2 (2022), pp. 123–133.

[59] Sumaira Ghazal, Arslan Munir, and Waqar S Qureshi. “Computer vision in smart agriculture and precision farming: Techniques and applications”. In: *Artificial Intelligence in Agriculture* 13 (2024), pp. 64–83.

[60] Mukesh Dalal and Payal Mittal. “A Systematic Review of Deep Learning-Based Object Detection in Agriculture: Methods, Challenges, and Future Directions.” In: *Computers, Materials & Continua* 84.1 (2025).

[61] Khan Zohaib, Yue Shen, and Hui Liu. “ObjectDetection in Agriculture: A Comprehensive Review of Methods, Applications, Challenges, and Future Directions”. In: *Agriculture* 15.13 (2025), p. 1351.

[62] Christos Charisis and Dimitrios Argyropoulos. “Deep learning-based instance segmentation architectures in agriculture: A review of the scopes and challenges”. In: *Smart Agricultural Technology* 8 (2024), p. 100448.

[63] Mrutyunjay Padhiary et al. “Enhancing precision agriculture: A comprehensive review of machine learning and AI vision applications in all-terrain vehicle for farm automation”. In: *Smart Agricultural Technology* 8 (2024), p. 100483.

[64] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (2002), pp. 2278–2324.

[65] Ross Girshick et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[66] Shaoqing Ren et al. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems* 28 (2015).

[67] Joseph Redmon et al. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[68] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. “Yolov4: Optimal speed and accuracy of object detection”. In: *arXiv preprint arXiv:2004.10934* (2020).

[69] Zhong-Qiu Zhao et al. “Object detection with deep learning: A review”. In: *IEEE transactions on neural networks and learning systems* 30.11 (2019), pp. 3212–3232.

[70] Anand Koirala et al. “Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of ‘MangoYOLO’”. In: *Precision Agriculture* 20.6 (2019), pp. 1107–1135.

[71] Inkyu Sa et al. “Deepfruits: A fruit detection system using deep neural networks”. In: *sensors* 16.8 (2016), p. 1222.

[72] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).

[73] Kai Han et al. “A survey on vision transformer”. In: *IEEE transactions on pattern analysis and machine intelligence* 45.1 (2022), pp. 87–110.

[74] Nicolas Carion et al. “End-to-end object detection with transformers”. In: *European conference on computer vision*. Springer. 2020, pp. 213–229.

[75] 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*. Springer. 2015, pp. 234–241.

[76] Shivam Chhabra and Rahul Rohilla. “A comparative study on semantic segmentation algorithms for autonomous driving vehicles”. In: *Ijraset J. Res. Appl. Sci. Eng. Technol.* (2022).

[77] 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*. 2015, pp. 3431–3440.

[78] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 7464–7475.

[79] Ning Guo et al. “Pose estimation and adaptable grasp configuration with point cloud registration and geometry understanding for fruit grasp planning”. In: *Computers and Electronics in Agriculture* 179 (Dec. 2020), p. 105818. issn: 0168-1699. doi: 10.1016/j.compag.2020.105818. url: https://www.sciencedirect.com/science/article/pii/S0168169920314046.

[80] Peteris Eizentals and Koichi Oka. “3D pose estimation of green pepper fruit for automated harvesting”. In: *Computers and Electronics in Agriculture* 128 (2016), pp. 127–140.

[81] Yatao Li et al. “Development and field evaluation of a robotic harvesting system for plucking high-quality tea”. In: *Computers and Electronics in Agriculture* 206 (2023), p. 107659.

[82] Hao Li et al. “Pose Estimation of Sweet Pepper through Symmetry Axis Detection”. en. In: *Sensors* 18.9 (Sept. 2018). Number: 9 Publisher: Multidisciplinary Digital Publishing Institute, p. 3083. issn: 1424-8220. doi: 10.3390/s18093083. url: https://www.mdpi.com/1424-8220/18/9/3083 (visited on 07/25/2025).

[83] Magni Hussain et al. “Green fruit segmentation and orientation estimation for robotic green fruit thinning of apples”. In: *Computers and Electronics in Agriculture* 207 (Apr. 2023), p. 107734. issn: 0168-1699. doi: 10.1016/j.compag.2023.107734. url: https://www.sciencedirect.com/science/article/pii/S0168169923001229 (visited on 07/17/2025).

[84] Fan Zhang et al. “Three-dimensional pose detection method based on keypoints detection network for tomato bunch”. In: *Computers and Electronics in Agriculture* 195 (Apr. 2022), p. 106824. issn: 0168-1699. doi: 10.1016/j.compag.2022.106824. url: https://www.sciencedirect.com/science/article/pii/S0168169922001417 (visited on 07/17/2025).

[85] Minho Jang and Youngbae Hwang. "Tomato pose estimation using the association of tomato body and sepal". In: *Computers and Electronics in Agriculture* 221 (June 2024), p. 108961. issn: 0168-1699. doi: 10.1016/j.compag.2024.108961. url: https://www.sciencedirect.com/science/article/pii/S0168169924003521 (visited on 07/25/2025).

[86] Guorui Zhao et al. "Selective fruit harvesting prediction and 6D pose estimation based on YOLOv7 multi-parameter recognition". In: *Computers and Electronics in Agriculture* 229 (Feb. 2025), p. 109815. issn: 0168-1699. doi: 10.1016/j.compag.2024.109815. url: https://www.sciencedirect.com/science/article/pii/S0168169924012067 (visited on 07/25/2025).

[87] Alessandra Tafuro et al. "Strawberry picking point localization ripeness and weight estimation". In: *2022 International Conference on Robotics and Automation (ICRA)*. May 2022, pp. 2295–2302. doi: 10.1109/ICRA46639.2022.9812303. url: https://ieeexplore.ieee.org/abstract/document/9812303 (visited on 07/17/2025).

[88] Mohd Ashaq Gaurav et al. "Robotics and Automation in Modern Agricultural Practices". In: (2024).

[89] Tom Duckett et al. "Agricultural robotics: the future of robotic agriculture". In: *arXiv preprint arXiv:1806.06762* (2018).

[90] Reza Rahmadian and Mahendra Widyartono. "Autonomous robotic in agriculture: a review". In: *2020 third international conference on vocational education and electrical engineering (ICVEE)*. IEEE. 2020, pp. 1–6.

[91] D Ramya et al. "Row guidance approach for navigation of semi-autonomous agriculture wheel robot". In: *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*. IEEE. 2019, pp. 1–4.

[92] Ailian Jiang and Tofael Ahamed. "Navigation of an autonomous spraying robot for orchard operations using LiDAR for tree trunk detection". In: *Sensors* 23.10 (2023), p. 4808.

[93] Yuseung Jo et al. "A Review on Dual-Arm Manipulation in Agriculture". In: *IEEE Access* (2025).

[94] Stavros G Vougioukas. "Agricultural robotics". In: *Annual review of control, robotics, and autonomous systems* 2.1 (2019), pp. 365–392.

[95] Tresna Dewi et al. "Visual servoing design and control for agriculture robot; a review". In: *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*. IEEE. 2018, pp. 57–62.

[96] Johannes F Elfferich et al. "Berrytwist: A twisting-tube soft robotic gripper for blackberry harvesting". In: *IEEE Robotics and Automation Letters* (2024).

[97] Christian Joel Lazo et al. “Automated Bell Pepper Quality Assessment: Robotic Gripper Sorting System with Transfer Learning”. In: *ECTI* 5.1 (2025), pp. 1–13.

[98] Soran Parsa, Bappaditya Debnath, Muhammad Arshad Khan, et al. “Autonomous strawberry picking robotic system (robofruit)”. In: *arXiv preprint arXiv:2301.03947* (2023).

[99] Pan Fan et al. “Three-finger grasp planning and experimental analysis of picking patterns for robotic apple harvesting”. In: *Computers and Electronics in Agriculture* 188 (2021), p. 106353.

[100] *L’insalata dell’Orto*. it-IT. url: `https://www.linsalatadellorto.it/` (visited on 08/26/2025).

[101] *Roboflow: Give your software the power to see objects in images and video*. en. url: `https://roboflow.com/` (visited on 08/26/2025).

[102] *ImageNet*. url: `https://www.image-net.org/` (visited on 08/26/2025).

[103] *Kaggle: Your Machine Learning and Data Science Community*. en. url: `https://www.kaggle.com/` (visited on 08/26/2025).

[104] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. doi: `10.1007/s11263-015-0816-y`.

[105] *Flowers-299*. en. Accessed: 2025-08-26.

[106] *Flower Color Images*. en. Accessed: 2025-08-26.

[107] Hugging Face. *OWL-ViT — Transformers documentation*. Accessed: 2025-08-26. url: `https://huggingface.co/docs/transformers/model_doc/owlvit`.

[108] Furkan TUNA. *Fruit Detection Dataset*. `https://universe.roboflow.com/furkan-tuna-vrcvi/fruit-detection-dp4ci`. Open Source Dataset. visited on 2024-08-19. Aug. 2023. url: `https://universe.roboflow.com/furkan-tuna-vrcvi/fruit-detection-dp4ci`.

[109] University of Tasmania. *Project 3 - Berries Dataset*. `https://universe.roboflow.com/university-of-tasmania-pzi8b/project-3-berries`. Open Source Dataset. visited on 2024-08-19. Oct. 2022. url: `https://universe.roboflow.com/university-of-tasmania-pzi8b/project-3-berries`.

[110] Eva Curto and Helder Araujo. “An experimental assessment of depth estimation in transparent and translucent scenes for Intel RealSense D415, SR305 and L515”. In: *Sensors* 22.19 (2022), p. 7378.

[111] Paul L Rosin et al. *RGB-D image analysis and processing*. Springer, 2019.

[112] Philip H Johnson et al. “Field-evaluated closed structure soft gripper enhances the shelf life of harvested blackberries”. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2024, pp. 9382–9388.

[113] DEFRA and Simon Pearson. *Automation in horticulture review*. Tech. rep. DEFRA, 2022, pp. 1–35. url: https://www.gov.uk/government/publications/defra-led-review-of-automation-in-horticulture/automation-in-horticulture-review.

[114] *NVIDIA Jetson TX2: High Performance AI at the Edge*. en-us. https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-tx2/. Accessed: 2025-08-27.

[115] *Jetson TX2 Module*. en-us. https://developer.nvidia.com/embedded/jetson-tx2. Accessed: 2025-08-27.

[116] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[117] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[118] Sinno Jialin Pan and Qiang Yang. “A survey on transfer learning”. In: *IEEE Transactions on knowledge and data engineering* 22.10 (2009), pp. 1345–1359.

[119] Walter F Wiggins and Ali S Tejani. “On the opportunities and risks of foundation models for natural language processing in radiology”. In: *Radiology: Artificial Intelligence* 4.4 (2022), e220119.

[120] Nusaybah M Alahdal et al. “Real-time object detection in autonomous vehicles with YOLO”. In: *Procedia Computer Science* 246 (2024), pp. 2792–2801.

[121] Niloofar Zendehdel, Haodong Chen, and Ming C Leu. “Real-time tool detection in smart manufacturing using You-Only-Look-Once (YOLO) v5”. In: *Manufacturing Letters* 35 (2023), pp. 1052–1059.

[122] Hongyu Zhao et al. “Real-time object detection and robotic manipulation for agriculture using a YOLO-based learning approach”. In: *2024 IEEE International Conference on Industrial Technology (ICIT)*. IEEE. 2024, pp. 1–6.

[123] Glenn Jocher. *Ultralytics YOLOv5*. Version 7.0. 2020. doi: 10.5281/zenodo.3908559. url: https://github.com/ultralytics/yolov5.

[124] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. *Ultralytics YOLOv8*. Version 8.0.0. 2023. url: https://github.com/ultralytics/ultralytics.

[125] Ali Farhadi and Joseph Redmon. “Yolov3: An incremental improvement”. In: *Computer vision and pattern recognition*. Vol. 1804. Springer Berlin/Heidelberg, Germany. 2018, pp. 1–6.

[126] Muhammad Hussain. "YOLOv1 to v8: Unveiling Each Variant–A Comprehensive Review of YOLO". In: *IEEE Access* 12 (2024), pp. 42816–42833. doi: 10.1109/ACCESS.2024.3378568.

[127] Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS". In: *Machine Learning and Knowledge Extraction* 5.4 (2023), pp. 1680–1716. issn: 2504-4990. doi: 10.3390/make5040083. url: https://www.mdpi.com/2504-4990/5/4/83.

[128] Chien-Yao Wang et al. "CSPNet: A new backbone that can enhance learning capability of CNN". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020, pp. 390–391.

[129] Kaiming He et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 37.9 (2015), pp. 1904–1916.

[130] Shu Liu et al. "Path aggregation network for instance segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8759–8768.

[131] Wei Liu, Irtiza Hasan, and Shengcai Liao. "Center and scale prediction: Anchor-free approach for pedestrian and face detection". In: *Pattern Recognition* 135 (2023), p. 109071.

[132] Mujadded Al Rabbani Alif and Muhammad Hussain. "YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain". In: *arXiv preprint arXiv:2406.10139* (2024).

[133] Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas". In: *Machine learning and knowledge extraction* 5.4 (2023), pp. 1680–1716.

[134] Herfandi Herfandi et al. "Real-time patient indoor health monitoring and location tracking with optical camera communications on the Internet of Medical Things". In: *Applied Sciences* 14.3 (2024), p. 1153.

[135] Muhammad Yaseen. "What is YOLOv9: An in-depth exploration of the internal features of the next-generation object detector". In: *arXiv preprint arXiv:2409.07813* (2024).

[136] Daniel Bolya et al. "Yolact: Real-time instance segmentation". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 9157–9166.

[137] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[138] Sixiao Zheng et al. “Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 6881–6890.

[139] Bowen Zhang et al. “Segvit: Semantic segmentation with plain vision transformers”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 4971–4982.

[140] Rishi Bommasani et al. *On the Opportunities and Risks of Foundation Models*. arXiv:2108.07258 [cs]. July 2022. doi: 10.48550/arXiv.2108.07258. url: http://arxiv.org/abs/2108.07258 (visited on 12/09/2023).

[141] R. Raja Subramanian et al. “FlowerBot: A Deep Learning aided Robotic Process to detect and pluck flowers”. In: Dec. 2022, pp. 1153–1157. doi: 10.1109/ICECA55336.2022.10009077.

[142] Bruno Siciliano et al. *Robotics: modelling, planning and control*. Springer, 2009.

[143] Yuanshen Zhao et al. “A review of key techniques of vision-based control for harvesting robot”. In: *Computers and Electronics in Agriculture* 127 (2016), pp. 311–323.

[144] Ning Guo et al. “Pose estimation and adaptable grasp configuration with point cloud registration and geometry understanding for fruit grasp planning”. In: *Computers and Electronics in Agriculture* 179 (2020), p. 105818.

[145] Justin Le Louëdec and Grzegorz Cielniak. “Key point-based orientation estimation of strawberries for robotic fruit picking”. In: *International Conference on Computer Vision Systems*. Springer. 2023, pp. 148–158.

[146] Hervé Abdi and Lynne J Williams. “Principal component analysis”. In: *Wiley interdisciplinary reviews: computational statistics* 2.4 (2010), pp. 433–459.

[147] MS Statistics Stat. “Principal component analysis”. In: (1987).

[148] Magni Hussain et al. “Green fruit segmentation and orientation estimation for robotic green fruit thinning of apples”. In: *Computers and Electronics in Agriculture* 207 (2023), p. 107734.

[149] Juan C Miranda et al. “Fruit sizing using AI: A review of methods and challenges”. In: *Postharvest Biology and Technology* 206 (2023), p. 112587.

[150] Ranjan Sapkota et al. “Immature green apple detection and sizing in commercial orchards using YOLOv8 and shape fitting techniques”. In: *IEEE Access* 12 (2024), pp. 43436–43452.

[151] Tao Li et al. “Occluded apple fruit detection and localization with a frustum-based point-cloud-processing approach for robotic harvesting”. In: *Remote Sensing* 14.3 (2022), p. 482.

[152] Van Nguyen Nguyen et al. “Templates for 3d object pose estimation revisited: Generalization to new objects and robustness to occlusions”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 6771–6780.

[153] Rama Bastola Neupane, Kan Li, and Tesfaye Fenta Boka. “A survey on deep 3D human pose estimation”. In: *Artificial Intelligence Review* 58.1 (2024), p. 24.

[154] Guozhao Shi, Fugui Zhang, and Xuemei Wu. “Robust keypoint-based method for peduncle pose estimation in unstructured environments”. In: *Computers and Electronics in Agriculture* 236 (2025), p. 110380.

[155] *scikit-learn: A set of python modules for machine learning and data mining*. `http://scikit-learn.org`. Accessed: 2024-01-15. (Visited on 01/15/2024).

[156] Yuita Arum Sari and Akio Gofuku. “Measuring food volume from RGB-Depth image with point cloud conversion method using geometrical approach and robust ellipsoid fitting algorithm”. In: *Journal of Food Engineering* 358 (2023), p. 111656.

[157] KM Lynch and FC Park. *Modern Robotics. Mechanics, Planning and Control*. Cambridge University Press, 2017.

[158] Francesco Amigoni et al. “Competitions for benchmarking: Task and functionality scoring complete performance assessment”. In: *IEEE Robotics & Automation Magazine* 22.3 (2015), pp. 53–61.

[159] Marcello Calisti et al. “Contest-driven soft-robotics boost: the robosoft grand challenge”. In: *Frontiers in Robotics and AI* 3 (2016), p. 55.

[160] Gonzalo Mier et al. “Fields2Benchmark: An open-source benchmark for coverage path planning methods in agriculture”. In: *Smart Agricultural Technology* (2025), p. 101156.

[161] Berk Calli et al. “The ycb object and model set: Towards common benchmarks for manipulation research”. In: *2015 international conference on advanced robotics (ICAR)*. IEEE. 2015, pp. 510–517.

[162] Obaloluwa Ogundairo et al. “Benchmarking Panoptic Segmentation Metrics that Reflect Downstream Robotic Manipulation Performance”. In: (Aug. 2025).

[163] Riccardo Bertoglio et al. “Quantitative Benchmarking in Agricultural Robotics”. In: *2020 I-RIM Conference*. I-RIM. 2020, pp. 1–2.

[164] Brian Yang et al. “REPLAB: A reproducible low-cost arm benchmark for robotic learning”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 8691–8697.

[165] Carlos Blanes Campos et al. “Technologies for robot grippers in pick and place operations for fresh fruits and vegetables”. In: *Spanish Journal of Agricultural Research* 9.4 (2011), pp. 1130–1141.

[166] Fabio Taddei Dalla Torre et al. “Artificial Intelligence Vision Methods for Robotic Harvesting of Edible Flowers”. In: *Plants* 13.22 (2024), p. 3197.

[167] Fabio Taddei Dalla Torre et al. “Toward autonomous blackberry harvesting with a soft gripper and vision-controlled robotic arm”. In: *2025 IEEE 8th International Conference on Soft Robotics (RoboSoft)*. IEEE. 2025, pp. 1–8.

La borsa di dottorato è stata cofinanziata con risorse dell'Unione europea-NextGeneration EU Piano Nazionale di Ripresa e Resilienza (PNRR)