ABC: Adaptive, Biomimetic, Configurable Robots for Smart Farms

From Cereal Phenotyping to Soft Fruit Harvesting

Fuli Wang

Supervisor: Dr Vishwanathan Mohan

School of Computer Science and Electronic Engineering

University of Essex

A thesis submitted for the degree of

Doctor of Philosophy

January 2023

Acknowledgements

“路漫漫其修远兮，吾将上下而求索”. This line comes from a long lyrical poem from the Chinese Warring States period. It expresses that there is still a long road ahead in the pursuit of truth, but the poet will spare no effort to explore it. During my PhD journey, I would like to thank Dr Vishwanathan Mohan, particularly, for exploring knowledge under scientific problems with a passion.

The first project I worked on during my PhD was developing code for shape identification in dense 3D point clouds. I am grateful to the project leader, Dr Richard Dudley from the National Physical Laboratory, who has improved my work forward by asking key questions. Additionally, my main work is a result of the past four years of the UK-China collaborative VERSATILE project funded by Innovate UK. I would like to thank Prof Dongbing Gu for his useful suggestions on the research and Mr Peter Hoyle, the Innovate UK Monitoring officer of the project, for valuable feedback on field trial results. I also extend my gratitude to Mr Andrey Ivanov and Mr Chris Newenham at Wilkin & Sons Ltd for providing access to the new vertical growing system for strawberries at Tiptree to run experiments as well as providing strawberry plants for the robotics laboratory.

I am proud to be doing research in the Agri-food robotics laboratory at the University of Essex and I would like to thank Mr Robin Dowling for providing me with a lot of technical support in terms of hardware and software. Thanks to my all colleagues and friends, namely Tao Chen, Leo Geer, Rodolfo Cuan Urquizo, Roberto Mendivil Castro, Penelope Roberts, and Omar Eldardeer. It has been a pleasure to work with you all.

Finally, I express my deep gratitude towards my lovely parents for their support and love.

Abstract

Currently, numerous factors, such as demographics, migration patterns, and economics, are leading to the critical labour shortage in low-skilled and physically demanding parts of agriculture. Thus, robotics can be developed for the agricultural sector to address these shortages. This study aims to develop an adaptive, biomimetic, and configurable modular robotics architecture that can be applied to multiple tasks (e.g., phenotyping, cutting, and picking), various crop varieties (e.g., wheat, strawberry, and tomato) and growing conditions. These robotic solutions cover the entire perception–action–decision-making loop targeting the phenotyping of cereals and harvesting fruits in a natural environment.

The primary contributions of this thesis are as follows. **a)** A high-throughput method for imaging field-grown wheat in three dimensions, along with an accompanying unsupervised measuring method for obtaining individual wheat spike data are presented. The unsupervised method analyses the 3D point cloud of each trial plot, containing hundreds of wheat spikes, and calculates the average size of the wheat spike and total spike volume per plot. Experimental results reveal that the proposed algorithm can effectively identify spikes from wheat crops and individual spikes. **b)** Unlike cereal, soft fruit is typically harvested by manual selection and picking. To enable robotic harvesting, the initial perception system uses conditional generative adversarial networks to identify ripe fruits using synthetic data. To determine whether the strawberry is surrounded by obstacles, a cluster complexity-based perception system is further developed to classify the harvesting complexity of ripe strawberries. **c)** Once the harvest-ready fruit is localised using point cloud data generated by a stereo camera, the platform's action system can coordinate the arm to reach/cut the stem using the passive motion paradigm framework, as inspired by studies on neural control of movement in the brain. Results from field trials for strawberry detection, reaching/cutting the stem of the fruit with a mean error of less than 3 mm, and extension to analysing complex canopy structures/bimanual coordination (searching/picking) are presented.

Although this thesis focuses on strawberry harvesting, ongoing research is heading toward adapting the architecture to other crops. The agricultural food industry remains a labour-intensive sector with a low margin, and cost- and time-efficiency business model. The concepts presented herein can serve as a reference for future agricultural robots that are adaptive, biomimetic, and configurable.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Robotics for Smart Agriculture – Why and why now

Interlinked factors, such as changing demographics, economics, and climate, are increasingly driving the trend toward using robotics in smart agriculture. Further, in July 2022, the Department for Environment Food & Rural Affairs (DEFRA, UK)-led review on automation in horticulture emphasised the need to accelerate/facilitate the adoption of artificial intelligence (AI)/Robotic harvesting technologies across horticulture. This can transform the 'low-tech' manual industry into a high-tech sector that contributes significantly to the gross domestic product (GDP), thus enabling food security through increased UK production, and minimising waste and carbon miles. The agricultural food industry is under severe pressure owing to the critical shortage of labour for tasks, such as fruit picking and packaging. Therefore, agricultural robots are being developed to both address the increased demand for production, and minimise production costs, and wastage, while ensuring environmental sustainability.

1.1.1 Economic drivers

An essential economic argument for using robots in agriculture is emphasising their potential to increase productivity and profitability through more efficient use of inputs [1]. For example, according to the British Summer Fruits (an industry body representing 95 % of UK-grown berries purchased by the UK's supermarkets and retailers) [2], the labour costs for producing strawberries/raspberries/blackberries/blueberries were £40,000–70,000 per hectare in 2020. The increasing demand for sustainable and cost-

effective growth year-round (52 w) in the UK can be satisfied only by scalability, workflow management, and robotic automation. In retail, "Big-shed" stakeholders, including Tesco, Sainsbury's, Morrison's, and Asda, are experiencing an increased demand due to the pandemic and Brexit. The fastest-growing sector in the UK is grocery online shopping, whereby Amazon Fresh, Mindful Chef (Nestle), Hello Fresh, and Gousto have gained popularity among consumers in recent years, which is expected to continue. Further, the demand for plant-based alternatives with excellent shelf life and provenance is growing. Most importantly, the current market climate has encouraged a focus on British producers and growers. The producers and growers must exploit their embedded robotics and automation to full capacity to meet market demands. The food market's volume is expected to reach 2.9250651 trillion kg by 2027, and the food market is expected to grow by 3.1 % in 2023 [3]. The UK food market constitutes only 2.5 % of the global food market by value; therefore, the export potential for smart robotic harvesting technology is approximately 40 times the UK market size. Productivity growth is critical to the economic sustainability of the economy, essentially highlighting the urgency to adopt robotic manufacturing solutions coupled with farm innovation.

1.1.2 Social drivers

Increased robot usage in agriculture is likely to impact the social fabric of rural communities in the long run [1]. First, the development and application of robotics in agriculture are expected to eventually decrease or eliminate some employment opportunities with low-skill requirements. Simultaneously, it may create new job opportunities, such as service engineers and remote operators. Additionally, as the global population continues to grow, several countries urgently need to address the low production yield in fruit and vegetable production, as well as the efficient and intelligent utilisation of resources. The technophilic promise of AI and robotics is expected to

displace existing agricultural labour hierarchies with a radical labour market: essentially, highly skilled, highly trained workers may use digital agricultural technologies to increase productivity and efficiencies, whereas, lower-skilled workers in the fields, greenhouses, processing plants, and warehouses may be subject to increased employer scrutiny and surveillance, further rationalisation of their workplaces, and increasing expectations of productivity [4]. Thus, although robotics can drive automation and digitisation in agriculture, it may also gradually convert low-skilled jobs into high-skilled jobs over the long term.

Regarding the current global workforce, optimism in the agricultural sector is currently lacking. The critical elements of the agricultural food industry are highly dependent on seasonal migrant labour to harvest crops. Recently, labour shortages were caused by economics and the Covid-19 pandemic, which severely affected the food and farming sector, whereby some fruit suppliers were forced to leave their produce rotting in the fields. This led agricultural practitioners to enquire about how, to what extent, and when can new robotic technologies and currently available automation ease the agricultural sector's dependency on seasonal labour.

However, the high production cost of fruit and vegetables might negatively impact affordability. Socially disadvantaged groups lack access to balanced nutrition, including fruits and vegetables, thus furthering inequality and the burden on the National Health Service. Cost-effective automation technologies can potentially improve yield and lower production costs, thereby enabling equal access to a balanced and nutritious diet.

1.1.3 Ecological drivers

Considerable research has reported that daily human activity is the cause of the rising global mean temperatures, which has resulted in current climatic conditions being the warmest in recorded history [5]–[7]. The AI for growing conditions can reduce energy,

which is a useful step toward achieving net zero. In particular, smart farms can effectively use resources and avoid waste. For example, improvements in crop handling accuracy, 24/7 operation, and control afforded by robotics can reduce waste and increase the rate of production, which is currently limited by the rate and efficiency at which manual operators can process produced times. The European Union 2021 [8] report stated that approximately 2 m tons of pesticides are used globally per year. Total fatalities worldwide resulting from unintended pesticide poisonings are estimated at approximately 11,000 deaths annually [9]. The automation of scouting, inspection, diagnosis, and treatment tasks via robots can significantly reduce the use of harmful pesticides and chemicals, thus improving yield and maintaining biodiversity.

Overall, developing robotics in smart agriculture may significantly increase production while being resource-efficient, being environment-friendly, mitigating prevalent labour shortages, minimising waste and carbon emissions, and being tolerant to climate variations; all of these are core challenges faced by the farming industry.

1.2 Central Contributions

Automation in Agri-food is an extreme case for handling a diverse range of produce, variations in the same type of produce, changing environmental conditions, and manipulation tasks involved. All existing automation solutions are expertly tailored to a specific product; nonetheless, functional recycling of the underlying perception, manipulation, and decision-making frameworks for versatility, reconfigurability, modularity, and adaptivity in the automated harvesting and smart farming processes presents tremendous scope.

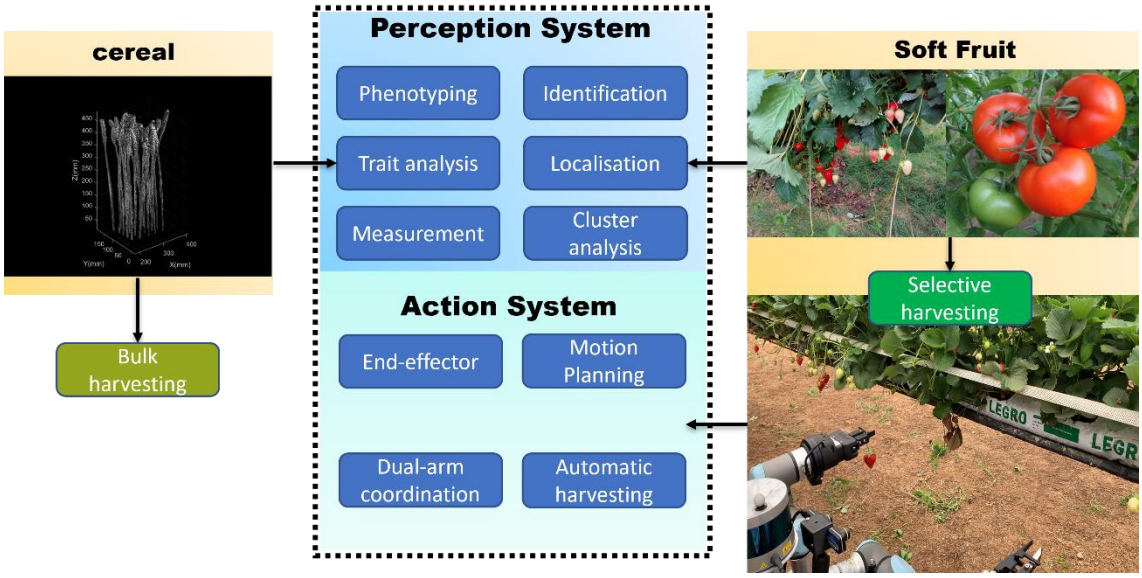


Figure 1. 1 - Robotic system developed by the thesis. The perception system provides targeted image-processing approaches for cereals and soft fruits, whereas the action system provides a solution for selectively harvested crops as cereals are typically left to the harvester.

In this study, a robotic perception–action system was developed for agricultural applications. Figure 1.1 briefly illustrates the proposed robotics system comprising perception and action parts. In the perception system, the phenotyping of cereals based on 3D point clouds and the RGB image classification/localisation of soft fruits are discussed respectively. To enable the robot to harvest the soft fruit, the action system is designed to control the dual-arm and mobile base for automatic harvesting. The contributions can be summarised as follows.

1. For the perception system, this thesis discussed two imaging processes, for wheat and strawberry. Accurate measurement of field-grown wheat traits, including spike number, dimension, and volume, are essential for crop phenotyping and yield analysis. Therefore, a high-throughput method for imaging field-grown wheat in three dimensions, along with an accompanying unsupervised measuring method for obtaining individual wheat spike data were presented. Images were captured using four structured light scanners on a field mobile platform, thereby creating dimensionally accurate sub-millimetre resolution 3D point clouds for a volume of 4.5 m^3 in less than 10 s. An adaptive

k-means algorithm with dynamic perspectives was used to fit each spike's shape and a random sample consensus algorithm was used to measure the dimensions. The method generates small cuboids to fit all the wheat spikes and estimate the total spikes volume.

2. In addition to cereals, conditional generative adversarial networks (GANs) trained using synthetic data, which was generated considering various environmental variances, were used. Compared with other models, this approach utilises the image-to-image translation technology to transform complex farm images into images containing only ripe strawberries; further, it eliminates the cumbersome manual data collection on farms and labelling. Herein, the recognition and localisation performance of the system was compared with human performance. Additionally, in real-world environments, some maturate strawberries are surrounded by stems and immature strawberries; to describe this type of situation, cluster complexity was defined. If no obstacles surround the target strawberry, this strawberry is classified as easy to harvest; otherwise, it is classified as difficult to harvest. To realise the classification of cluster complexity, a YOLACT-based model was developed by training the images from a greenhouse. This thesis compared and analysed the two models. These models could be integrated into our Essex agricultural robot (EAR).

3. High variability in the canopy structure of the crop, occlusions, and minimising damage owing to contact impose a range of task-specific constraints for the robot action system. For robot manipulation actions, this study developed a novel neural control framework for goal-directed reaching considering various task constraints (e.g., gripper pose, joint limits, timing, bimanual coordination, and alignment of the gripper/cutter to the stem). The action system is a

forward/inverse model that can be used to simulate the consequences of actions for predictive planning and an extension to a range of tools coupled to the arm.

4. The perception–action system was implemented on the Essex agricultural robot. In addition to the experiments in the laboratory setting, field trials were conducted with the robot in the UK's first new vertical growing system for soft fruit at Tiptree, Essex, within the framework of an Innovate UK Industrial Strategy Challenge Fund on Transforming Food Production program (UK-China) through the project 'Versatile-Configurable, Smart Indoor harvesting of 'Aubergine, Tomato and Strawberry' crops (Project ID- 107460, 2021–2023).

1.3 Organisation of This Thesis

Developing a commercially viable automated harvesting and precision farming solution for an indoor, controlled environment utilising new sensors, and a data-driven crop management approach to maximise yield and minimise waste and emissions remains a challenge. Therefore, this study attempted to develop a versatile and configurable perception–action system for robotic harvesting. The overall framework is illustrated in Figure 1.2. The perception and action systems are presented separately, and then the two systems are integrated and detailed laboratory and field trials are given.

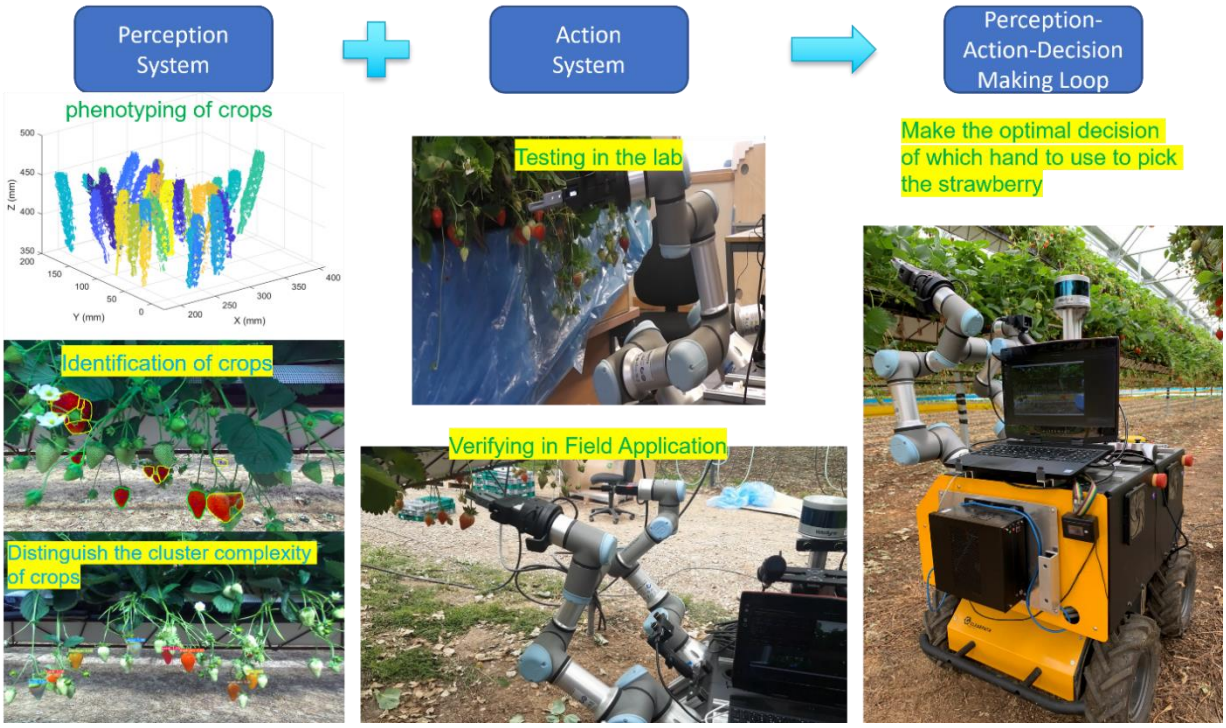


Figure 1. 2 - Overall framework of the thesis: The perception system comprises functions to handle wheat crop traits analysis and crop identification. Whereas the action system is integrated with the perception system into the Essex agricultural robot, thereby enabling the robot to harvest fruit automatically.

The rest of this thesis is organised as follows. Chapter 2 presents the background on robotic perception–action. With developing hardware (e.g., stereo camera, Lidar, and light scanners) and algorithms (e.g., machine learning), the perception system has significantly improved in both crop identification and phenotyping. However, for the action system, employing robotic arms in unstructured environments is currently challenging, particularly for bimanual processes, such as picking a ripe strawberry with one hand while moving the surrounding unripe strawberries away with the other. Understanding the differences between current bimanual robot control approaches and how the human brain considers two-handed manipulations might reveal factors contributing to the bimanual manipulation ability gap between humans and robots [10]. Therefore, in addition to the works from optimal control theory (OCT), this chapter provides related work on impedance control based on the equilibrium point hypothesis

and synergy formation. Subsequently, some recently developed crop-harvesting robots are reviewed, and finally, the contribution of this thesis is summarised.

Chapters 3 and 4 focus on the crop's measurement and identification. The hardware equipment and algorithms to measure or identify crops may vary with the crop. This thesis considered the soft fruit strawberry as the primary object for robot harvesting; nonetheless, it differs from some cereals in that cereals do not exhibit noticeable feature changes in shape and colour upon ripening. For, example, when wheat gradually matures, the volume of the wheat spikes increases accordingly; however, its colour does not change. These cereals can be harvested with large harvesters; however, yield estimates require advanced machine vision methods to replace manual measurements. Therefore, these two chapters independently study wheat dimension measurement and strawberry identification.

In chapter 5, the action system for the robot is developed. It is a neural network implementation of the passive motion paradigm (PMP) based on the impedance control and equilibrium point hypothesis. This chapter explains and analyses the theoretical model of PMP and some experimental results.

In Chapter 6, to ensure that the robot can continue harvesting ripe strawberries on the farm, a perception–action loop system is built. The robot should recognise ripe strawberries in the entire system and devise a forward plan for all detected strawberries. Particularly, it should decide which strawberry is suitable for harvesting by left, or right arm or combined with the mobile base movement. The proposed robotic perception–action system is verified in field applications. Further improvements and open questions are discussed at the end of the chapter.

Chapter 7 presents the general conclusions and some brief ideas for future work.

Finally, as this thesis is centred on agricultural robots and graphics cannot fully demonstrate the performance of the proposed system, videos of some tests in

experimental and on-farm environments are provided in the chapter on Supplementary Material.

1.3 Summary of Achievements

The following conference papers have been published during my PhD study.

- **F. Wang**, V. Mohan, A. Thompson and R. Dudley. “Dimension fitting of wheat spikes in dense 3D point clouds based on the adaptive *k*-means algorithm with dynamic perspectives,” 2020 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), 2020, pp. 144-148.

- L. Geer, D. Gu, **F. Wang**, V. Mohan and R. Dowling, "Novel Software Architecture for an Autonomous Agricultural Robotic Fruit Harvesting System," 2022 27th International Conference on Automation and Computing (ICAC), 2022, pp. 1-6, DOI: 10.1109/ICAC55051.2022.9911161.

The following journal papers have been published during my PhD study.

- **F. Wang**, F. Li, V. Mohan, R. Dudley, D. Gu, and R. Bryant, “An unsupervised automatic measurement of wheat spike dimensions in dense 3D point clouds for field application,” Biosyst. Eng., vol. 223, pp. 103–114, 2022, DOI: https://doi.org/10.1016/j.biosystemseng.2021.11.022.

- **F. Wang**, R, Rodolfo Cuan, P. Roberts, et al. Biologically inspired robotic perception-action for soft fruit harvesting in vertical growing environments. Precision Agric 24, 1072–1096 (2023).

Finally, during my PhD, I also attended the following workshop.

Plant Feature Extraction from 3D Point Clouds Workshop, PhenomUK workshop, 1st July 2021. (Oral presentation)

Chapter 2

Background on Robotic Perception–Action for Crop Harvesting

This chapter provides the necessary background and suggestions for further research on the current state-of-the-art perception/vision systems, action systems (motion control methods), and prototypes of agricultural robots.

2.1 Crop Perception System Based on Machine Learning

Recently, several interesting vision approaches for tackling this challenge have been proposed. The literature on crop recognition technology is particularly extensive. For apples, strawberries, and tomatoes, the obtained RGB image is typically input into the recognition model (classifier), and subsequently, each harvest-ready crop is identified for the robot to harvest. The difference is that cereals, such as barley and wheat plants, typically use image-based plant phenotyping to measure and analyse trait variations, whereas harvesting tasks are handled directly by larger harvesters. Therefore, this section presents the different image processing methods for fruits and cereals.

2.1.1 Image-based classifiers

Machine learning classifiers, including both supervised and unsupervised learning, have recently become a popular method. Mathematically, the dataset can be denoted as $D = \left\{ x^i \right\}_{i=1}^{N}$, consisting of N examples. The difference between supervised and unsupervised learning is that supervised learning requires each instance to be labelled.

Therefore, the dataset for the supervised learning method can be denoted as $D = \left\{ x^i, y_j^i \right\}_{i=1}^{N}$. Where each instance x^i is associated with a label $y_j^i \in [y_1, ..., y_L]$, and L elements exist corresponding to L label concepts. The classifier can provide predictions $\hat{\mathbf{y}} = \left[\hat{y}^i, ..., \hat{y}^N \right]$ for a given dataset.

Different classifiers have varying advantages and disadvantages. For other types and sizes of data, choosing an appropriate model and pre-processing operations and setting the parameters are often complex.

Regarding the application of crop recognition, various recognition algorithms, such as colour-based analysis, edge detection, *k*-means clustering, and Bayes classifications, have been provided and discussed (see [11] and the references therein). These methods use the obtained images as data, use feature extraction methods for pre-processing, and subsequently input them into the corresponding algorithms for processing. For example, a method based on a histogram of oriented gradients (HOG) descriptor associated with a support vector machine (SVM) classifier was proposed for detecting strawberries [12]. Furthermore, another study extracted and combined tomatoes' shape, texture, and colour features to achieve accurate tomato recognition based on the SVM [13]. The SVM is based on the principle of structural risk, and minimisation can minimise the upper bound on expected risk and implement classification using a separating hyperplane determined by a few support vectors. Therefore, SVM is less prone to overfitting or local optimal solutions compared with other methods, and it can be generalised for small sample sizes [14]. However, the detection performance of this type of method is influenced by the feature extraction methods and parameter selection.

Artificial neural network (ANN)-based (deep learning) object detection has recently attracted significant attention owing to its powerful learning ability and advantages in handling occlusion, scale transformation, and background switches [15]. Unlike in the traditional methods, in ANN-based methods, the input data is initially forwarded to a

feature extraction network, and subsequently, the resultant extracted features are forwarded to a classifier network [16]. Therefore, both feature extraction and classification can be performed by neural networks without data pre-processing. Given the above advantages, several ANNs have been introduced to detect fruits or vegetables for harvesting robots. For instance, the mask region convolutional neural network (Mask-RCNN) [17] was introduced into the machine vision of a strawberry harvesting robot for fruit detection; thus, it improved universality and robustness in a non-structural environment [18]. Convolutional neural networks (CNNs) have also been developed to detect, segment, and track wine grapes [19]. Additionally, a vision system to localise strawberries based on the Mask-RCNN has been developed [20]; this system aims to avoid collisions between the gripper and fixed obstacles. However, the localisation algorithm still needs to optimise and adapt to suit more complex situations, such as occluded and unusual hanging positions of the strawberries. In addition to strawberries, a rich image dataset of date fruit bunches in an orchard comprising over 8,000 images of five date types in different pre-maturity and maturity stages has been created and tested [21]. Furthermore, a team at the University of Cambridge [22] initially trained Vegebot to recognise the harvest-ready, immature, infected lettuce, and background in the field using the YOLOv3 [23]. Overall, with the development of ANNs, object recognition performance has improved significantly over the past decade.

2.1.2 Imaged-based plant phenotyping

In addition to recognising crops, measuring and analysing trait variations in crops over various seasons is crucial, particularly for cereals. Accurate and repeatable trait measurement is essential for success in phenotyping applications. For example, major phenotypes for wheat breeding are the number of spikes, spike length/width, and volume. Several techniques have been explored for collecting data for quantitative studies of

complex traits related to growth, yield, and adaptation to biotic or abiotic stress [24]. Spike counting is one of the main approaches used for predicting grain yield in cereals [25]. To count the number of wheat spikes, a simple particle count algorithm on segmented 2D images was developed [26]; however, it failed to address the high crop density and overlapping spikes. Reducing count errors in dense, close contact spikes was explored [27] using an automatic spike-counting algorithm and zenithal colour 2D images of the crop in natural light. Algorithms, such as DeepCount and YOLOv5 [28]–[30], have been developed to count the number of wheat spikes in 2D images using deep convolutional neural networks and machine learning approaches. Achieving volumetric or dimensional information is challenging, particularly when captured from directly above or at an angle where distortions are introduced and thus, partial visibility masks the real size of the spike. Calibration charts can mitigate distortions and mosaicking errors; however, they are complex to implement for high-throughput field studies.

Generating a 3D, digital twin of a cereal plot offers a significantly richer and more dimensional correct representation, thus overcoming issues of obscured and overlapping spikes. A digital twin can be generated by combining multiple 2D images or utilising more complex imaging technology, such as Lidar, time-of-flight, and structured light scanners [31]. Thus, the field captured data is no longer represented by a 2D RGB image but rather by a 3D point cloud, with format Pn (x, y, z, RGB). Algorithms used for 2D image analysis are no longer applicable for point clouds and alternative approaches have been developed using supervised neural networks to fit complex geometric primitives, such as CAD models of mechanical components [32], [33]. In this task of wheat phenotyping, there are some simple geometric primitives involved. Therefore, a more classical clustering algorithm can be used for segmenting the wheat and subsequently fitting it to spikes; thus, the process of training a supervised model can be omitted, and

the fitting results can be obtained faster. For instance, [34] performed wheat spike segmentation using two different classical methods: voxel-based and mean shift segmentation. Additionally, the density-based spatial clustering of applications with noise (DBSCAN) algorithm [35] was developed for segmentation, and least-squares curve fitting was used to obtain the size of the wheat spikes [36]. Although the clustering algorithms, such as DBSCAN, mean shift, and *k*-means, can be successful in segmentation tasks, the segmentation task can be challenging for these algorithms in specific complex environments, such as when wheat crops are extremely dense. Admittedly, using existing measurement algorithms to obtain a robust measurement result with less time duration and computing resources is still challenging.

2.2 Robot Motion Control for Harvesting Actions

Generally, two types of harvesting approaches are implemented by agricultural practitioners to reduce orchard labour expenses: selective and bulk harvesting [37]. Bulk harvesting uses large harvesters to harvest cereals or vibrating tree trunks to harvest fruit, whereas selective harvesting involves humans or robots selectively picking ripe crops.

Selective harvesting is a more complex harvesting method for robotic systems utilising manipulators with end-effectors for picking. Therefore, industrial research has focused primarily on the manipulation and end-effector in robots. For instance, several control schemes of grippers for harvesting crops were designed in laboratory environments [38]–[40]; however, no field experiments were conducted to verify their performance on farms. Another study [41] presented the design and field testing of a robotic system designed to harvest apples. The harvesting system successfully picked 127 out of a total of 150 fruits, thus achieving an overall success rate of 84 %. However, the picking rate of more fragile soft fruits must be improved while ensuring that the crops are not damaged. The cherry harvesting robot developed in Japan consists of a 4-degree-of-

freedom (DoF) manipulator, 3D vision sensor, and end-effector [42]. Given the nature of the cherry tree, the team created an articulated manipulator with an axis that moves up and down and three axes that turn left and right. However, experiments revealed that the manipulation action may damage the target fruit if other fruits surround it. The end-effector is equipped with soft rubber components; however, this is not always effective. One of the central issues in control movement is the DoF problem. Essentially, the same movement goal can be reached by an infinite number of combinations of the control variables. This is the well-known inverse kinematic problem of determining a vector of joint variables that produce the desired end-effector location. However, the inverse kinematics problem can be ill-posed because either no solution exists (in this case the target location is infeasible, i.e., out of the reachable workspace) or multiple solutions exist [43]. Optimal and impedance controls have played key roles in overcoming this problem. This section presents the development of and related work on robot motion control based on the two methods.

2.2.1 Methods based on optimal control theory

The general idea of robot motion control based on the OCT involves determining an optimal control scheme from a class of allowable control variables. This can be achieved by defining an objective function (cost function). From a mathematical perspective, it can be expressed as: under the constraints of the equation of motion and the allowable control variables, the extreme value of the objective function (minimum value of the cost function) is obtained. In 1985, Flash and Hogan [44] presented the objective function as the square of the magnitude of the jerk of the hand integrated over the entire movement. The solution of such a minimisation task was consistent with the spatial–temporal variances reported by [45]. Subsequently, an extensive literature of similar studies involving varying objective functions, such as integrated torque change [46], minimum object crackle [47]

and minimum acceleration criterion [48], emerged. Recent developments indicate that OCT has gradually emerged as a robust theory for interpreting a range of motor behaviours [49], online movement corrections [50] and structure of motor variability [51]. Furthermore, to compute the optimal solution of the manipulator, several inverse kinematics algorithms have been implemented in dedicated motion planning software [52].

However, a fundamental challenge in this approach is deriving the optimal control signal for a nonlinear time-varying system, given a specific objective function and assumptions regarding the noise structure [53]. The mathematics of computing an optimal feedback controller is extremely complex [54]. Additionally, becoming stranded in local optimum is a widespread problem in optimisation algorithms. The nonlinear optimisation framework for inverse kinematics requires further exploration.

2.2.2 Methods based on impedance control

Between the mid-1960s and mid-1980s, several neuroscience studies proposed the equilibrium point hypothesis (EPH) [55]–[57] to explain neural control of movement. The basic idea of this hypothesis is that posture is not directly controlled by the brain in a detailed manner but rather is a biomechanical consequence of equilibrium among a large set of muscular and environmental forces. Several studies [58]–[62] using intact and spinal cord animals revealed that muscle synergies may construct motor behaviours with the associated force fields organised within the brain stem and spinal cord, and activated by descending commands from supraspinal areas.

For mechanical manipulators, an alternative to OCT is impedance control, which was developed by considering the mechanics of the interaction between physical systems [63]. Given that manipulation is a fundamentally nonlinear problem, the distinction

between impedance and admittance is essential, and as the environment contains inertial objects, the manipulator must be an impedance.

For the impedance control of manipulators, mathematically, the actuator is assumed to generate the commanded torque $\mathbf{T}$ with the actuator angle, θ, and a kinematic relationship between the actuator angle and end-point (end-effector) exists such that $\mathbf{x} = L(\theta)$. Designing a feedback control law that coordinates the desired relation between end-point force $\mathbf{F}$ and position $\mathbf{x}$ for implementing in an actuator is quite straightforward. To define the desired equilibrium position for the end-point without environmental forces as $\mathbf{x_0}$, a general form for the desired force–position relation is: $\mathbf{F} = K(\mathbf{x_0} - \mathbf{x})$. According to the Jacobi matrix $J(\theta)$ and principle of virtual work, the required relation in actuator coordinates is $\mathbf{T} = J^T(\theta) K(\mathbf{x_0} - L(\theta))$.

The relation $K(\mathbf{x_0} - \mathbf{x})$ does not present any linear restrictions. The relation selected to make the end-point stiff accomplishes Cartesian end-point position control and eliminates the inverse kinematics problem; only the forward kinematic equations for the manipulator must be computed.

To shift the cost function in OCT to the force field in impedance control, a neural network implementation of the PMP [64] has been developed for robot manipulation based on the EPH [53], [65]. Qualitatively, the process by which the brain determines the distribution of work across a redundant set of joints when the end-effector is assigned the task of reaching a target point in space can be represented as an "internal simulation process" that calculates how much each joint would move if an externally induced force (i.e., the goal) pulls the end-effector by a small amount toward the target. The mechanism labelled "passive" aligns with the EPH because the brain does not explicitly specify the equilibrium point; instead, it contributes to the activation of "task-related" force fields.

[66] detailed this novel perspective of viewing motor control and summarised the principle of a neural network implementation of the PMP.

Recently, PMP has been applied in different contexts, such as combining postural and focal synergies during whole-body reaching tasks [67], and coordination of the movements of the upper body of the iCub along with the paintbrush to derive motor commands for drawing the shapes [68], [69]. Regarding the application of the agricultural robot, this thesis applied the PMP for goal-directed reaching considering various task constraints (e.g., gripper pose, joint limits, timing, bimanual coordination, and alignment of the gripper/cutter to the stem). The action system is a forward/inverse model that can simulate the consequences of actions for predictive planning and extend to a range of tools coupled to the arm.

2.3 Survey of State-of-the-art Robotic Harvesters

With the rapid development of computer vision, artificial intelligence, and robotics control, several robotics systems and prototypes have been developed for crop harvesting, both in the research and commercial fields.

2.3.1 Robotic harvesters in the research phase

Some of the recent research literature regarding tomato, strawberry, sweet pepper and lettuce crop-harvesting robots is as follows.

Tomato.

A dual-arm robot was developed for harvesting tomatoes in a greenhouse [70]. However, the DoF of this type of double manipulator is limited; it has some restrictions under uncertain conditions. To improve the success rate, optimal sorting and fruit nearest neighbour positioning algorithms were developed for determining the position of the tomato fruit and estimating the grasping pose [71]. Based on the optimised picking

strategy, the manipulator's harvesting success rate was 72.1 %. Additionally, 6D pose (3D translation + 3D rotation) estimation of maturity-classified tomatoes was developed to assist the robot to detect the stem accurately for the harvesting process [72].

Figure 2. 1- Tomato robotic harvesters developed in [70]–[72].

Strawberry.

[73]–[75] discussed 3D location methods of the vision systems and presented an autonomous strawberry - harvesting system, which had a gripper at the end of the manipulator to pick strawberries. However, the gripper was not sufficiently dextrous and contacted/damaged the harvest-ready and immature strawberries simultaneously. In addition, a co-robotic harvest-aid system and its evaluation during commercial strawberry harvesting were developed to improve harvesting efficiency [76]. However, this system is designed to aid picking staff; hence, it cannot handle the autonomous harvesting of strawberries.

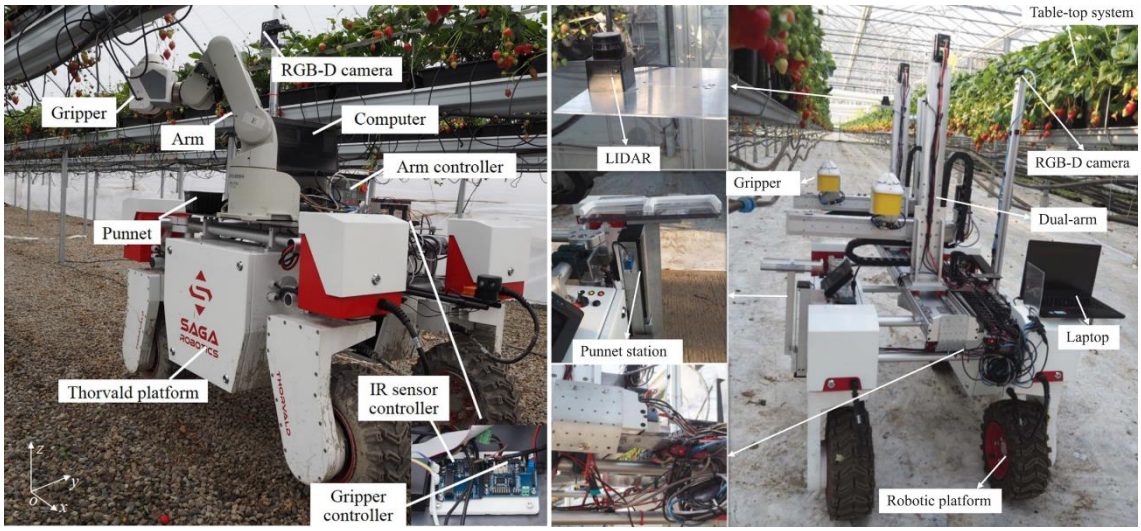


Figure 2. 2 - Strawberries robotic harvesters developed in [73], [74].

Sweet pepper.

A few robotics platforms for harvesting sweet pepper fruit in a greenhouse have been proposed to improve the performance in commercial greenhouses [77], [78]. However, the success rate of crop harvest needs to be improved compared with those of human workers. To prevent possible collision damage in the near-neighbour multi-target picking of sweet peppers by robots in densely planted complex orchards, [79] proposed an algorithm for recognising sweet peppers and planning a picking sequence. Although this study presented a method that can localise sweet peppers in a densely planted environment, the picking performance lacks field testing.

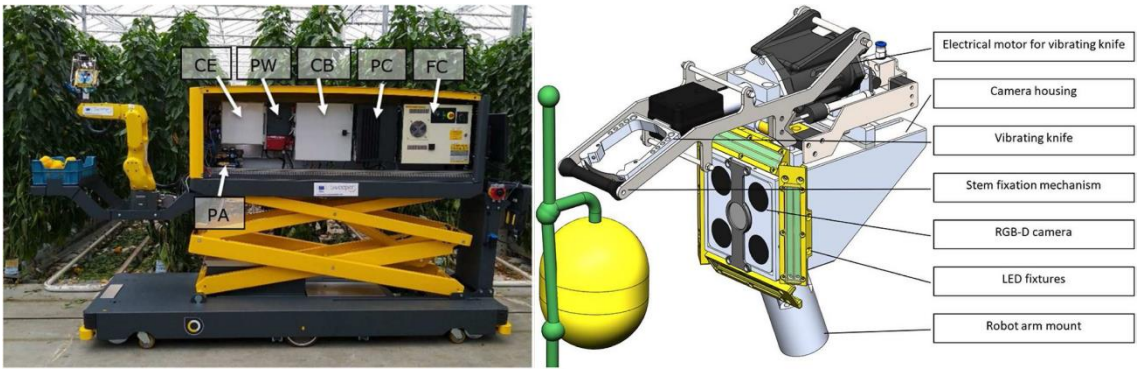


Figure 2. 3 - Sweet pepper harvester with its end-effector [78].

Lettuce and cabbage.

DeepLabV3+ model, a deep learning technology, was used to segment abnormal leaves for hydroponic lettuce sorting [80]. The Vegebot platform [15] developed a custom end-effector and software to harvest iceberg lettuce; however, it is not yet suitable for commercialisation. As shown below in Figure 2.4, [81] proposed a backstepping control-based attitude control system for cutting devices to harvest cabbage. However, this harvester is integrated with a driver platform, which requires a human operator.

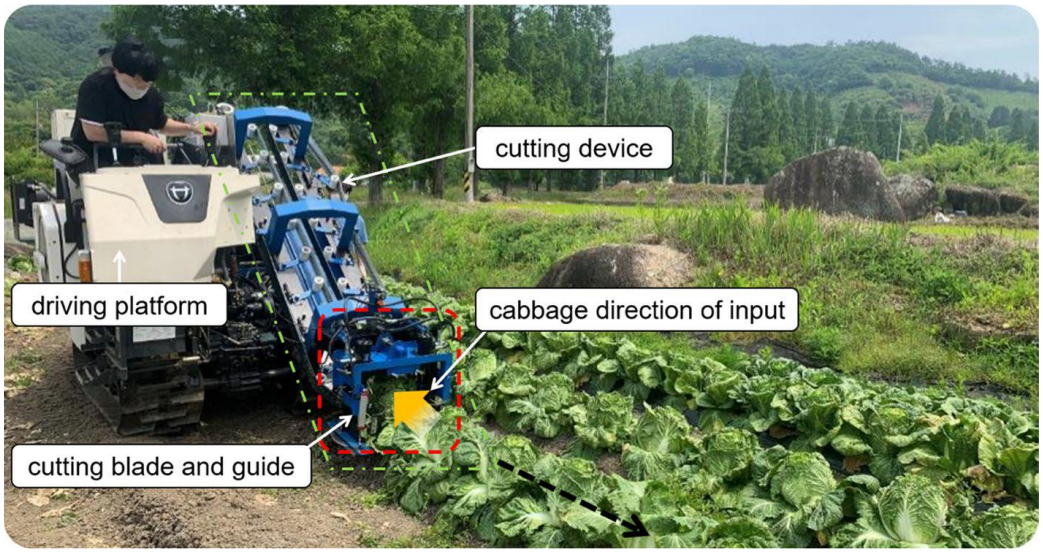


Figure 2. 4 - Cabbage harvester with the driver platform and cutting device [81].

2.3.2 Commercially available systems

In addition to the value of academic research, these developments present some commercial prospects. For example, the Shadow Robot company build next-generation robot hands and systems with advanced dexterity to help push the state-of-the-art in dexterous manipulation. The dexterous humanoid robot hands are reliable for object handover [82] and might be helpful in fruit harvesting. Additionally, several companies are already developing and producing independent modular robots or other related technologies to provide agricultural services. These include Octinion, an innovative R&D company specialised in mechatronic product development applied to biological material, and Thorvald, which is committed to developing autonomous modular robots that can be

configured for most agricultural environments. Furthermore, a new robot is being developed by Fieldwork Robotics, a spin-out company from Plymouth University, and it can enable farmers to pick over 25,000 raspberries daily. Finally, the literature on research and commercial agricultural robots for use in crop field operations has been reviewed [83] and concludes that current agricultural robotic systems still need to improve the robot's hand-eye coordination.

2.4 Beyond the State-of-the-art: Essex Agricultural Robot

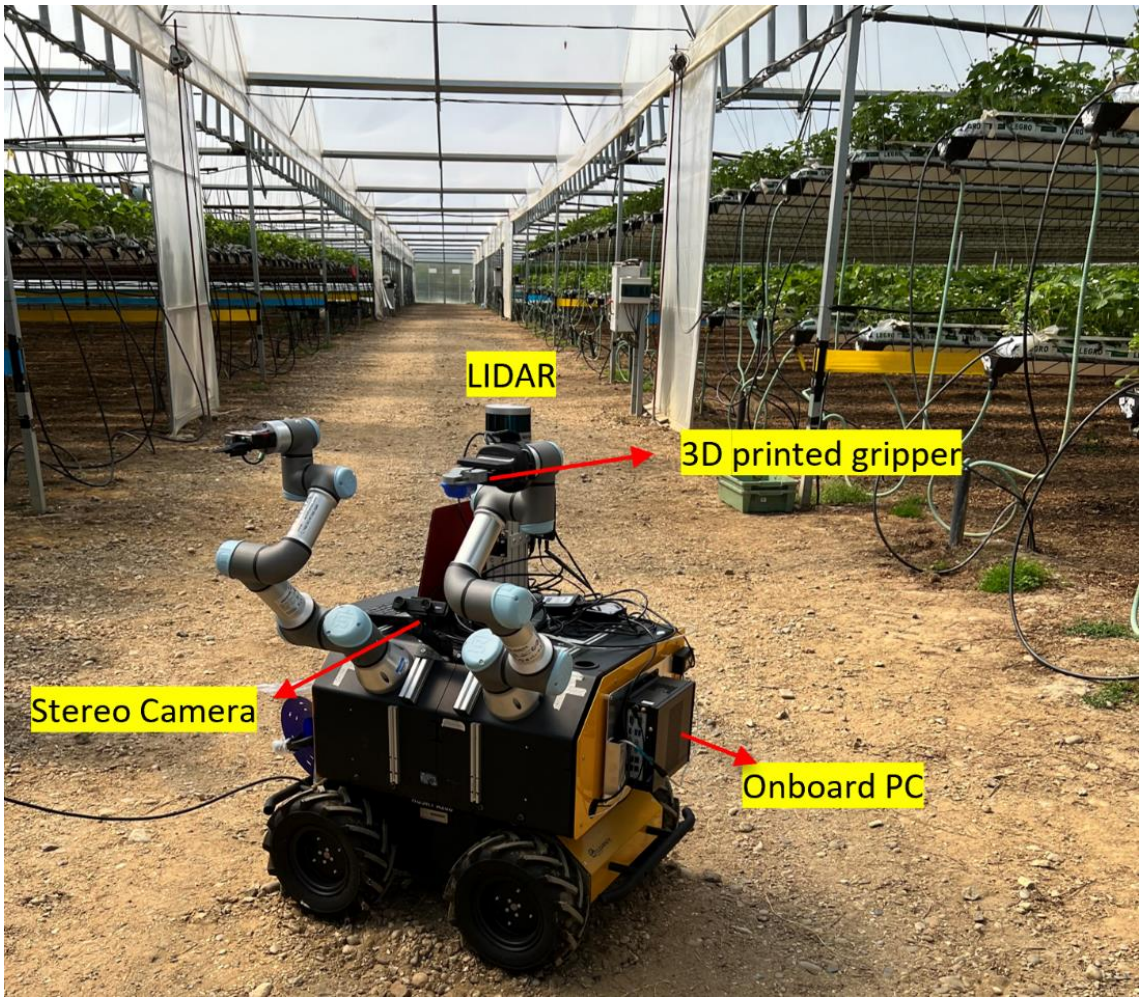


Figure 2. 5 - Essex agricultural robot.

As shown in Figure 2.5, the EAR comprises a mobile vehicle with two 6-degree-of-freedom (DoF) universal robots, 3D printed gripper/cutters, and a range of sensing capabilities (ZED stereo camera and Lidar); further, it is powered by a rechargeable

lithium iron phosphate battery. Because the current technologies designed for agricultural robots can be improved, this thesis developed a configurable perception–action system and applied it to the EAR for field application.

Chapter 3

Configurable Crop Perception I: Phenotyping of Cereal-Wheat

In the following two chapters, two image processing methods are presented for handling cereals and fruit separately. Herein, an unsupervised automatic measurement of wheat spike dimensions in dense 3D point clouds was proposed for cereals, such as the wheat plant. Regarding the fruit, supervised neural networks used to detect and localise the strawberries were developed. The details regarding the two image process methods are described as follows.

Traditional manual measurement of wheat spikes' sizes is usually done by random sampling a unit square meter of the wheat field and measuring it by using a ruler. As shown in Figure 3.1, there are hundreds of wheat plants per unit square metre, manual measuring the wheat spikes' size is a laborious task on the field. Therefore, a core challenge is phenotyping cereal in the field to replace the manual measurement method.

Figure 3. 1 - Wheat field picture demonstrates the laborious task of measuring wheat spikes.

3.1 Wheat Dimensions Measurement via 3D Point Clouds

Image-based plant phenotyping is a rapidly emerging research area that can provide quantitative measurement of the structural and functional properties of plants for the development of new plant varieties. However, the trait analysis and disease detection of wheat plants are primarily conducted manually by human experts using a process [84]. Because the manual measurement is laborious, the dimensions of wheat spikes must be measured using imaging methodologies instead to enable high-throughput phenotyping.

The maturity of wheat can be evaluated only by fitting the spike size, which does not involve numerous complex geometric primitives. To collect the dataset, three different 3D imaging technologies have been compared in reference [31]: multi-stereo imaging, time-of-flight and structured light laser scanning to produce point clouds of a wheat plant in situ. In this work, the method of light scanners is used to capture the 3D point cloud of crops and clustering algorithms to separate spikes from wheat crops. Clustering algorithms, such as the DBSCAN and *k*-means algorithms, are well suited to the task; admittedly, some defects still exist when dealing with practical situations.

The *k*-means algorithm can be described as: given a set of *n* samples $\{x_1, x_2, ..., x_n\}$ and a positive value *k*. The algorithm aims to partition these *n* samples into *k* clusters by minimising the distortion, which is the within-cluster sum of the distances from each sample to its nearest centroid. The key idea of DBSCAN is that for each sample of a cluster, the neighbourhood of a given radius (*Eps*) must contain at least a minimum number of neighbours (*MinPts*), which implies that the cardinality of the neighbourhood must exceed a certain threshold. One of the disadvantages of the DBSCAN is that its performance depends considerably on the parameters selected (*Eps* and *MinPts*), but they lack a theoretical basis. Therefore, the trial method is commonly used, it relies predominantly on experience, which results in the final parameters not necessarily being

optimal [85]. Instead, the *k*-means algorithm has the characteristics of a single parameter, and its parameter *k* represents the cluster number.

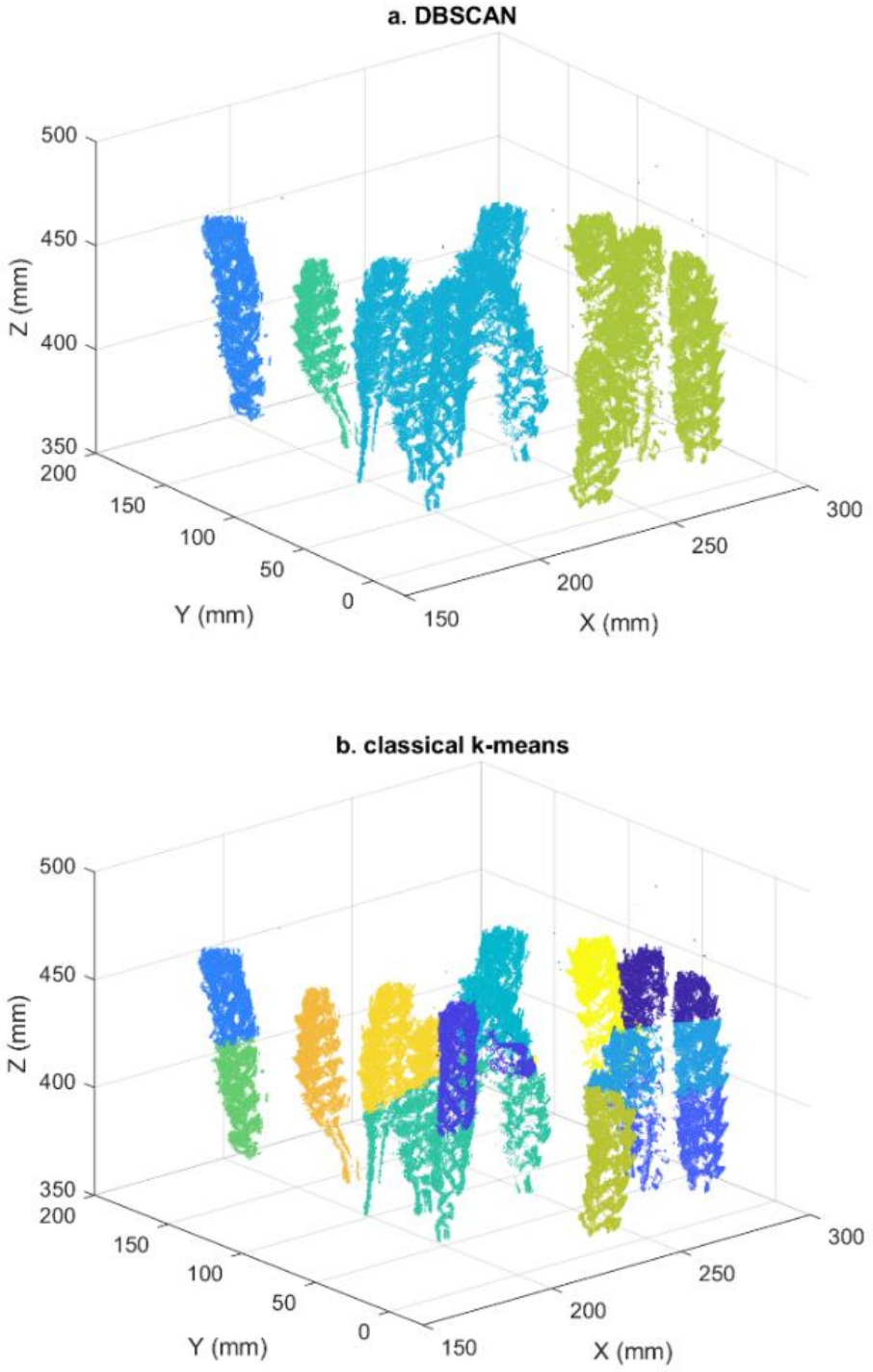


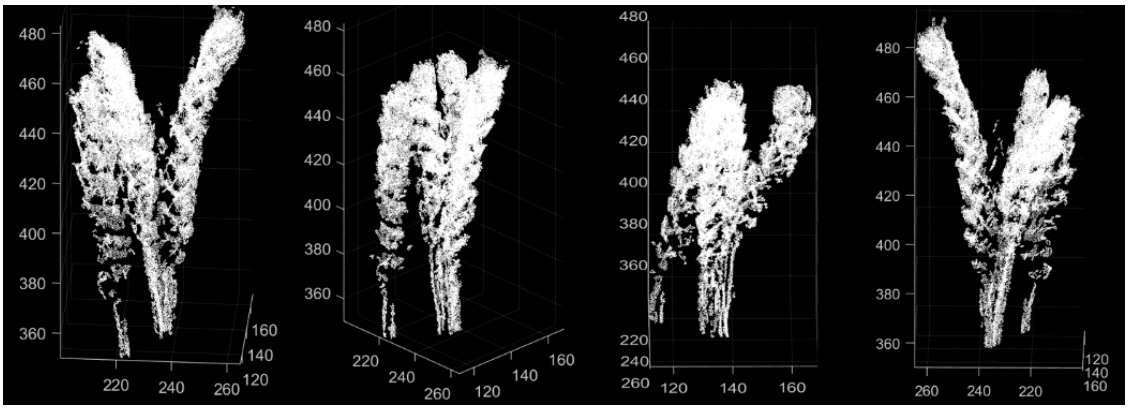
Figure 3. 2 - Results of DBSCAN and classical *k*-means segmentation. The DBSCAN cannot identify every individual spike. The classical *k*-means divides one spike into multiple segments.

To compare these two classical algorithms, Figure 3.2 demonstrates the segmentation results of the DBSCAN (here, *MinPts* is set as ten, and *Eps* as five) and classical *k*-means (*k* = 12). As the number of spikes was 12, the parameter of *k*-means was set as 12 and the trial method was used to set the relatively reasonable parameters of the DBSCAN. The DBSCAN algorithm divided the 12 spikes into four segments; essentially, DBSCAN identified only four spikes (with different colours in Figure 3.2 a), which is not able to cluster all individual spikes. Meanwhile, in the classical *k*-means algorithm, even when the number of clusters was set to 12, the output result was poor. These results illustrate that the classical clustering algorithm cannot handle complex environments, such as when wheat crops are considerably dense.

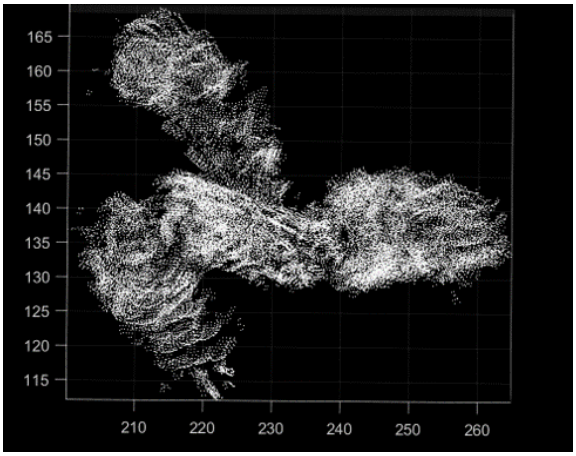
3.2 Adaptive *k*-means Algorithm for Wheat Dimensions Measurement

To address the above concerns, an adaptive *k*-means algorithm with dynamic perspectives was proposed herein. As shown in Figure 3.32, when wheat crops were observed from the side, the wheat spike and stem could be easily distinguished. Owing to overlapping between the spikes, the number of spikes could not be easily evaluated from the side view (Figure 3.3 a). However, the top view could be used to count the number of spikes (Figure 3.3 b). Similarly, for the *k*-means algorithm, if all of the 3D points are projected into the 2D top view, the point distance in the within-cluster is reduced and the clustering performance is improved.

One view can reflect only the form of an object in one orientation, not the complete structural shape of the object. This implies that more information can be obtained if perspective to see things is changed.



a) Side views



b) Top view

Figure 3. 3 - Spikes observation with different perspectives (unit: mm). The side view clearly shows some wheat stalks and wheat spikes, but not necessarily the number of wheat plants. By contrast, the top view indicates the number of wheat spikes present; however, no stalk is visible.

To improve segmentation performance, the above idea was introduced into the *k*-means algorithm. The flowchart of the *k*-means algorithm with dynamic perspectives is shown in Figure 3.4. In particular, given a cluster consisting of points $N_{n\times 3}$ (Figure 3.4 a), where *n* is the number of points, and three is the number of dimensions, $\{x_i\}$, $\{y_i\}$ and $\{z_i\}$ denote the x, y and z coordinates of the point $i(i \in n)$. For the side view, the $N_{n\times 3}$ array was transferred into an $N_{n\times 2}$ array, which contains only the two dimensions of $\{x_i\}$ and $\{z_i\}$. The 2D points were inputted from the side view into the *k*-means, which outputs all point labels. Using the labels to mark all 3D points, the clustering result in Figure 3.4 b was achieved. To separate spikes from the wheat (Figure

3.4 c), Algorithm 3.1 was defined to preserve the top segments. Similarly, by transferring the 3D points of spikes into the top view $N'_{n\times 2}$, which contains only two dimensions of $\{x_i\}$ and $\{y_i\}$, the segmentation result was obtained based on the top view in Figure 3.4 d; evidently, the result was superior to those of the classical algorithms. Finally, a random sample consensus (RANSAC) algorithm [86] was used to fit each segment shape and obtain the dimensions as shown in Figure 3.4 e.

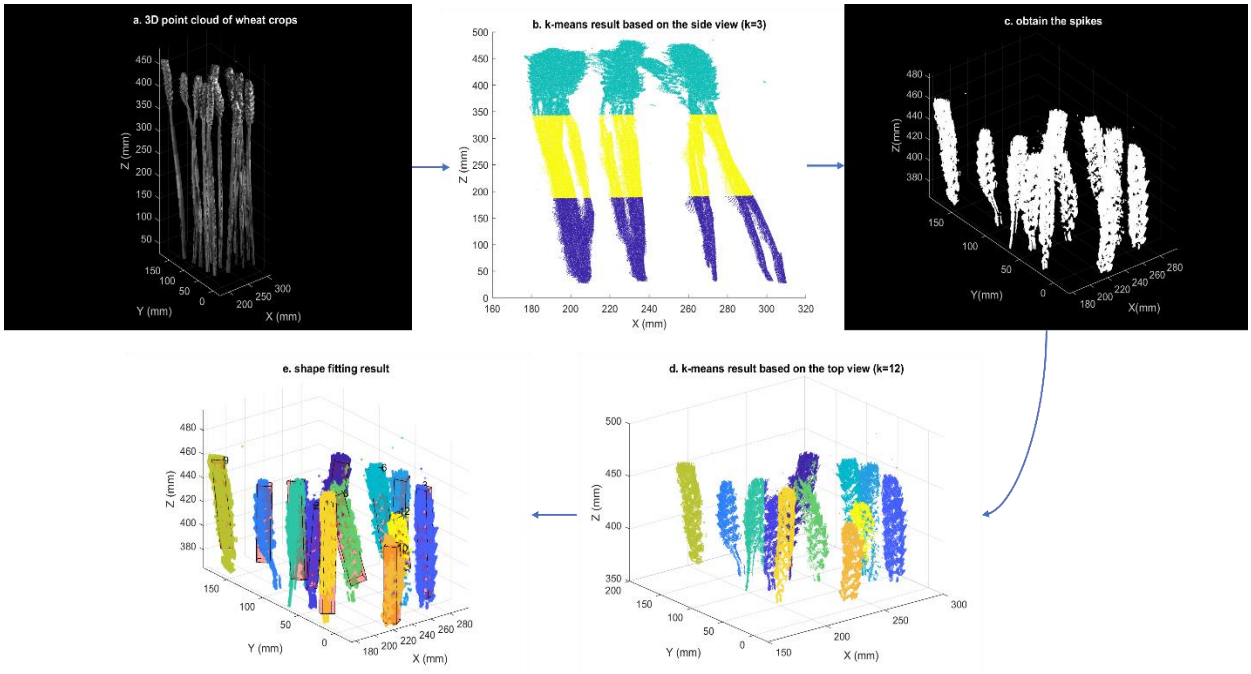


Figure 3. 4 - Segmentation results based on the proposed *k*-means. The proposed method first segments the original image (a→b) to obtain the wheat spikes' part (c); next, each wheat spike is separated/identified (d); finally, a shape fit is performed for each wheat spike to estimate the size.

In Algorithm 3.1 defined below, a value, which has the max value of $\{z\}$, was defined according to the highest point of all 3D points. By extracting all the segments in the value space, the points belonging to spikes were obtained. In Figure 3.5, the highlighted area is the value space determined by the parameter σ. Once this value space was defined, to preserve the top segments, information regarding whether the highest point of each segment was located in this space was required. For the first image of Figure 3.5, owing to the small parameter value, the part of the green cluster is not located in the

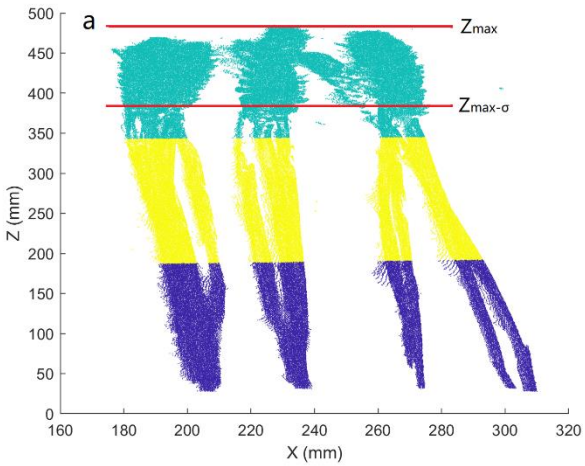
highlighted area, whereas the highest point is in it. Therefore, for the small parameter, if the highest point of the cluster is located in its space, the cluster is considered as wheat spikes and retained. The second image corresponds to the most perfect parameter value, and this case is less frequent. For a larger parameter (the last picture), both green and yellow clusters' points are located somewhere in the space. To retain only the cluster of the wheat spikes, the decision condition must be changed to whether the lowest point of the cluster is located in the space. In this study, the conditional statement (with σ set to 60 mm) was set to evaluate if the highest point is located in the value space, and statistical filtering was used to reduce the noise.

Algorithm 3.1: Obtaining wheat spikes

Require: 3D points: $N_{n\times 3}$;
Initialize parameter of σ;
Reduce the noises of 3D points;
Obtain side view 2D points: $N_{n\times 3} \rightarrow N_{n\times 2}$
Use the *k*-means for segmentation based on side view;
Obtain the point with the highest *Z* coordinate value: $Z_{\max}$;
Calculate a value space of *Z* coordinates:

$$\left[Z_{\max} - \sigma, Z_{\max}\right]$$

For each highest point within its segment:
    If the highest point is located in the value space: Preserve this segment *// the decision condition*
    else: continue;
return all preserved segments.



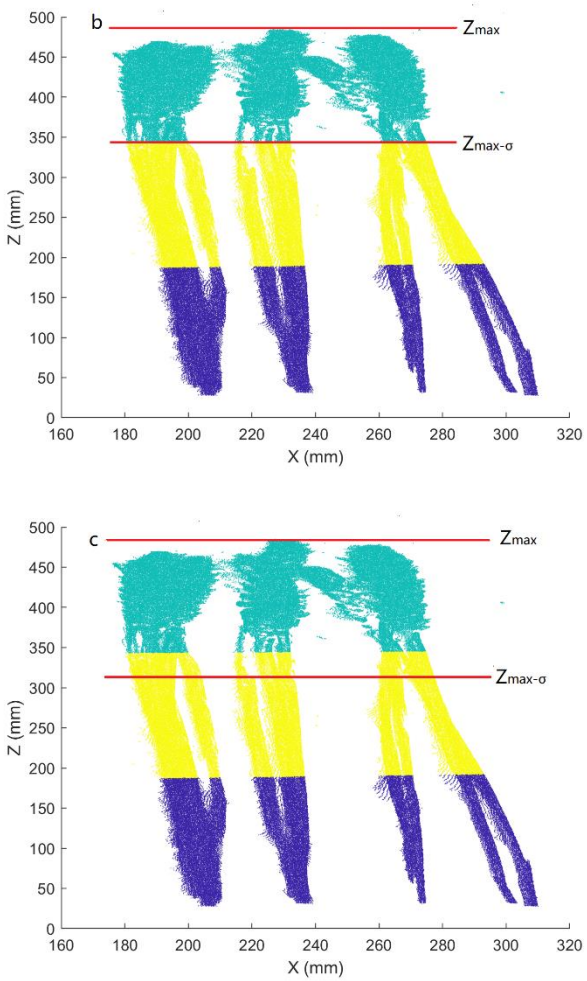


Figure 3. 5 – Three different value spaces for spikes obtaining. Using Algorithm 3.1 to preserve the top segments, the space value need not be accurately set; ensuring that σ is a small value (figure a and b are both the correct spaces that can output the same result). If the conditional statement in the algorithm is changed to evaluate whether the lowest point of each segment is located in this space, the σ would be set to a larger value (figure c and b would be the correct spaces).

As previously discussed, setting the *k*-means parameter to three or four for the side view is sufficient. Because the shape of the wheat crop is similar to a cuboid or cylinder, selecting the side view from the X or Y direction can achieve the same spike's height and width results. Further, shape fitting for each spike is required with the number of clusters, that is, the number of spikes, set in advance. However, the number of wheat spikes may not be known in advance in the real world. Thus, the algorithm must calculate the number of spikes. To realise this function, an adaptive operation to self-update the appropriate

parameter values was added. The detail of this adaptive *k*-means algorithm based on dynamic perspectives is described in Algorithm 3.2.

Algorithm 3.2: Adaptive *k*-means algorithm based on dynamic perspectives

Obtain the wheat spikes according to Algorithm 3.1
Set the initial parameter k' for the top view
Obtain top view 2D points: $N'_{n\times 3} \rightarrow N'_{n\times 2}$
repeat
  Use the *k*-means for segmentation based on the top view;
  Use RANSAC to fit each segment;
  Evaluate the size of each segment;
  if (there is an abnormal size)
    $k'++$;
    break;
  end if
until there is no abnormal size
return the updated shape model.

An initial parameter k' is required to perform the segmentation for the top view in the algorithm. The value of this initial parameter should be small to ensure that the algorithm can update it adaptively. After obtaining the initial parameter k', the algorithm uses *k*-means to segment the spikes based on the top view and subsequently calls the RANSAC algorithm to fit a cuboid to each segmentation. Because the initial value k' is small, the fitting result is inaccurate. As shown in Figure 3.6 a, when the k' is small, some abnormal spike sizes are outputted (the fitting size of the purple part is significantly larger than that of regular wheat). Therefore, once the algorithm detects unreasonable results, k' is superimposed until a reasonable final result is outputted (Figure 3.6 b), which means that for each loop, the value of k' is added by 1. The last updated k' value is the number of spikes counted by the algorithm.

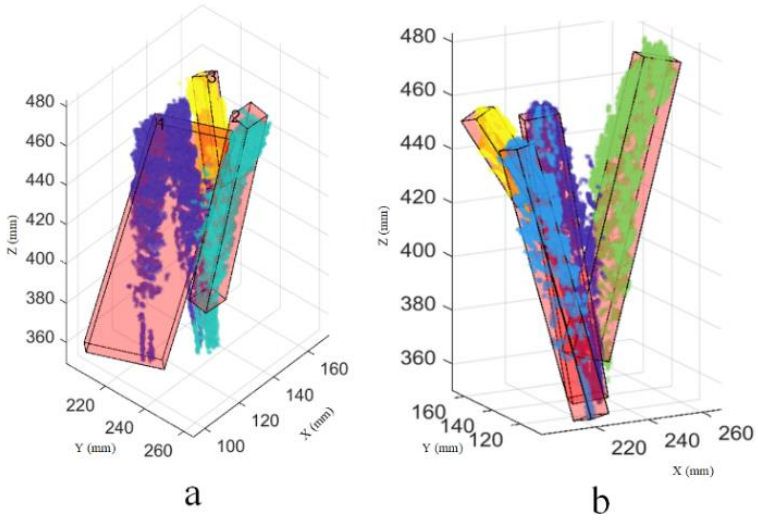


Figure 3. 6 - (a) Shape fitting result with abnormal sizes ($k' = 3$); here, two spikes are fitted by one cuboid. (b) Final shape-fitting result ($k' = 4$); here, each spike is properly fitted.

The algorithm did not make any intrinsic change to the *k*-means algorithm; however, it required several iterations of any existing implementation of *k*-means. Therefore, the algorithm can call any version of the *k*-means. Considering the computational performance, Lite *k*-means [87] or ball *k*-means [88] are recommended to run the proposed algorithm.

3.3 Framework of the Proposed Method for Wheat Field Application

Although Algorithm 3.2 can handle the environment where multiple wheat spikes are grown densely better than classical algorithms, directly applying the algorithm with images captured over a wide area is challenging. For example, as shown in Figure 3.7, compared with sample wheat crops in the laboratory, the captured 3D point cloud images from the field have hundreds of spikes and may contain noise; thus, existing measurement algorithms may not be able to obtain a robust measurement result. If the algorithm is directly used for handling these images captured from the field, the computational efficiency is significantly reduced as the images contain considerable noise and multiple

wheat crops. In addition, noise interference makes it difficult to output ideal results without abnormal dimensions.

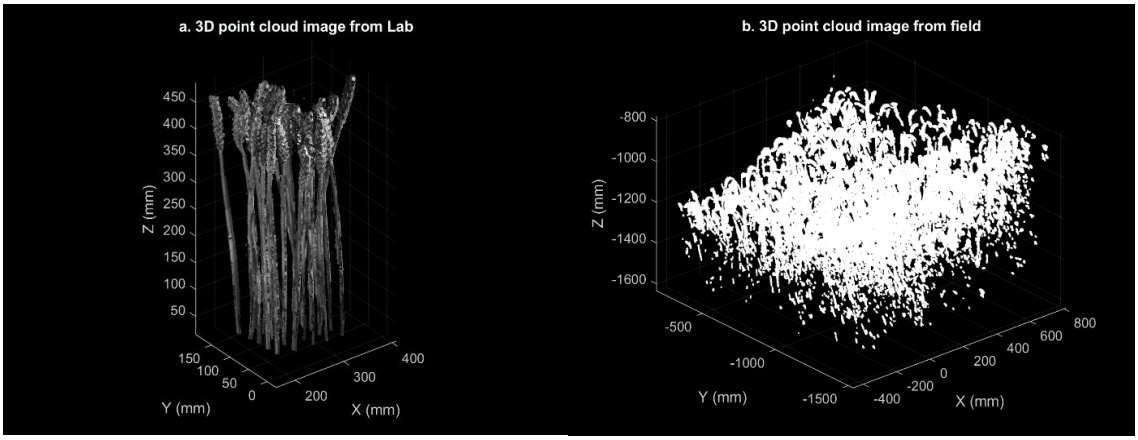


Figure 3. 7 – Dense 3D point cloud images of wheat crops. The picture from the laboratory clearly shows each wheat plant, whereas the image scanned from the wheat field contains hundreds of wheat plants and a large amount of noise.

To address the above problem, a method was proposed to extend Algorithm 3.2, as shown in Figure 3.8. First, the original field image was divided into a few segments and some stems were removed. As shown in Figure 3.8, the original image was divided into three segments; next, the spikes volume of each segment was individually calculated. To compute the volume, 3,000 small cuboids were used to fit the shape of all spikes for each segment thus, the volume of spikes was the sum of the volumes of all cuboids. After the volume calculation, some small areas were selected as sample areas (red highlighted areas in Figure 3.8), and then Algorithm 3.2 was used to calculate the average size of these areas. Overall, for images from the wheat field, the total spikes volume and average size of a single spike could be estimated by the proposed method.

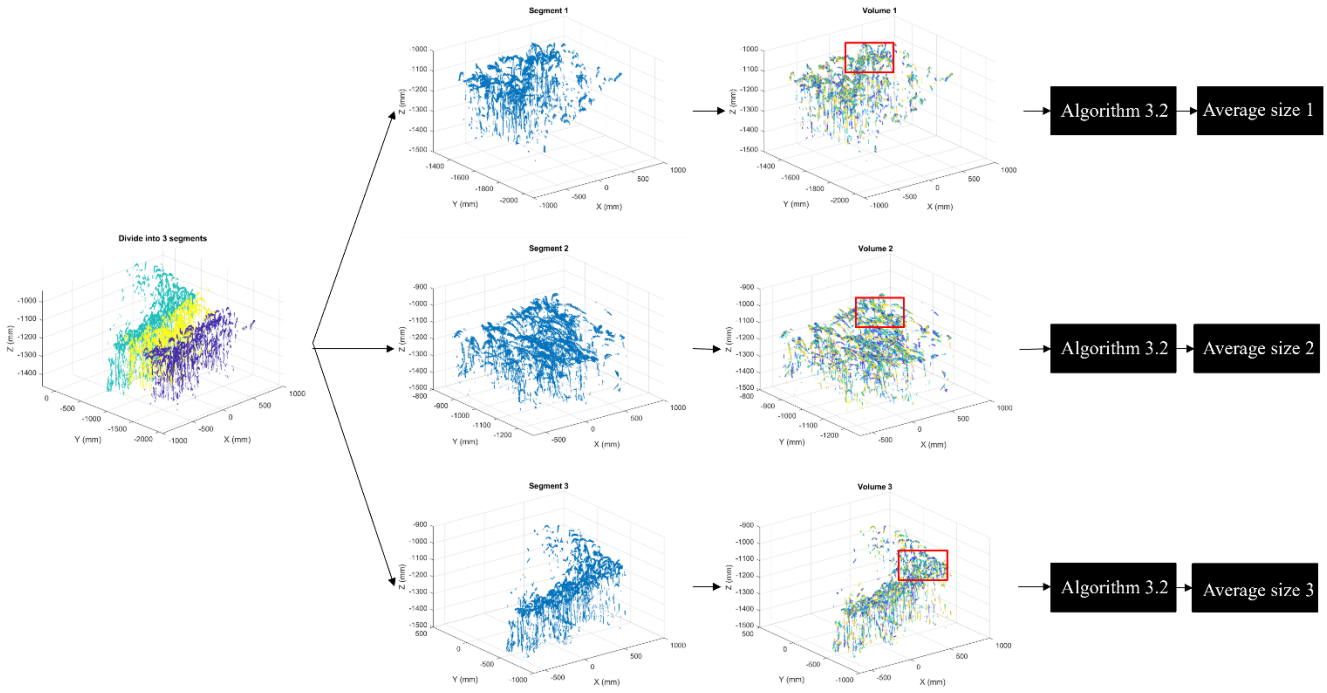


Figure 3. 8 - Overall flowchart of the proposed measurement method of wheat spikes. As the plot contains a large amount of wheat, the original picture is split into three images for separate processing. Here, 3,000 small cubes are employed for each image to fit/estimate the total volume. Next, some sample regions are extracted and the proposed algorithm is used to estimate the average size of the spikes in the sample regions.

The above description involves two parameters, namely, the number of segments and the number of small cuboids. In this study, all of the original images were divided into three segments and 3,000 cuboids were employed for each segment. If the value of these parameters is increased, the accuracy of the calculation results may improve, along with an increase in calculation cost, which is undesirable.

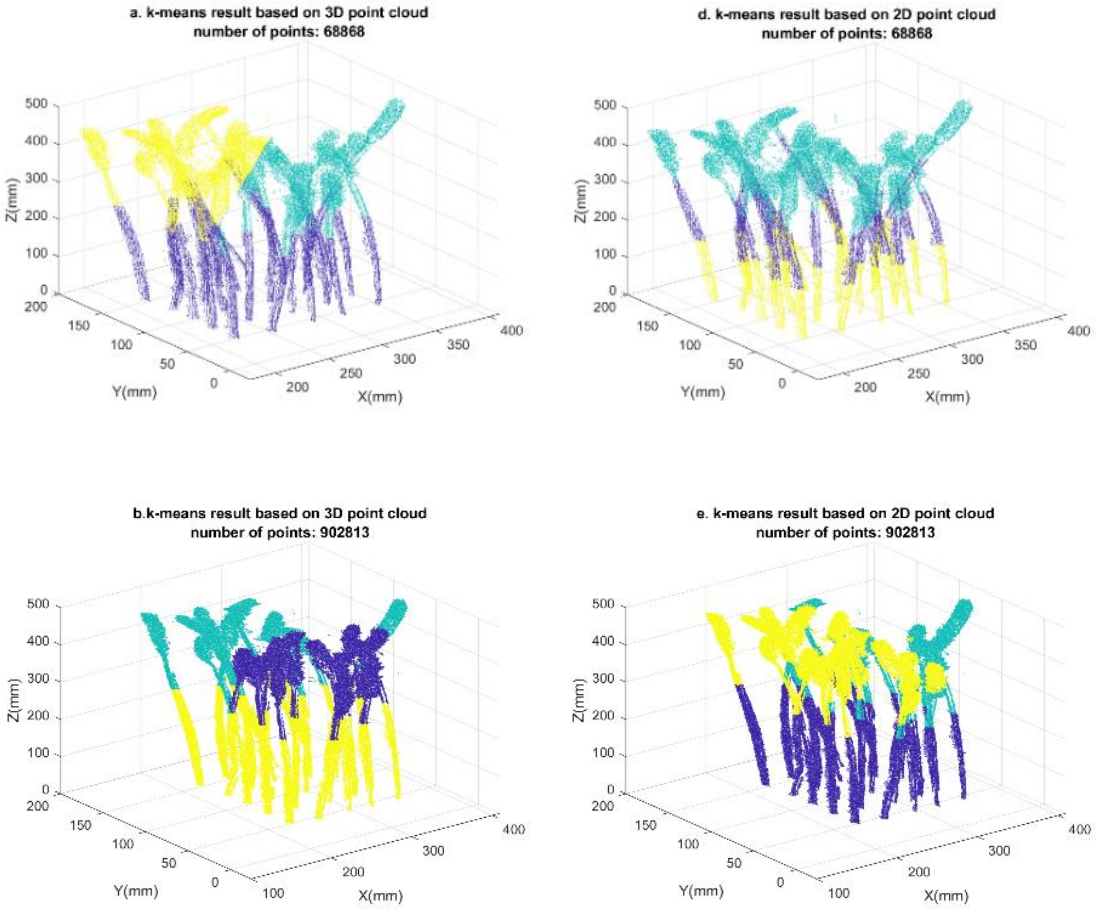
3.4 Experimental Analysis and Field Trials of Wheat Dimensions Measurement

3.4.1 Analysis of the proposed *k*-means algorithm

Before field trials, the performance of Algorithm 3.2 must be analysed as it is a core algorithm in the proposed method. The proposed *k*-means algorithm is a two-stage

method. In the first phase (Algorithm 3.1), the 3D point cloud image is projected into a 2D point cloud (side view); this is a dimension reduction process. To verify if this dimension reduction can output good results and improve the speed of the algorithm, some experiments were conducted and the results are shown in Figure 3.9.

As shown in Figure 3.9, for the same scene of the 3D point cloud, the number of points was adjusted by down-sampling. The algorithm was run five times to calculate the average results that were implemented in MATLAB R2020b based on a Core i9-9980HK CPU 2.40 GHz laptop. The comparison of running time between the 3D and 2D point clouds is presented in Table 3.1.



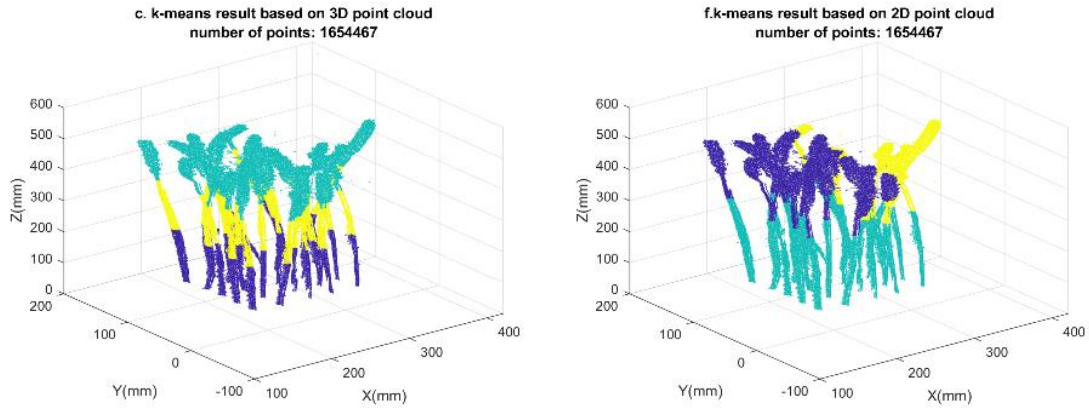


Figure 3. 9 - Results of *k*-means based on 3D and 2D point clouds. Note that the *k*-means assigns clusters to each point. After obtaining the output results, the different clusters' points are uniformly labelled on the 3D image using different colours.

Table 3. 1 – Comparison of running time between 3D and 2D point clouds

Number of points	Running time of 3D point cloud	Running time of 2D point cloud
68868	1.91 s	1.67 s
902813	9.28 s	8.69 s
1654467	16.86 s	15.12 s

Evidently from Table 3.1, using *k*-means to process 3D and 2D point cloud images, the results obtained were similar; however, with an increase in points, the computational efficiency of the 2D point cloud improved. The running time included the entire time, from loading the point cloud to drawing the resulting picture. Additionally, the results obtained using Lite *k*-means were compared with those of the traditional *k*-means; evidently, the Lite *k*-means process significantly improved the calculation speed using the operation mechanism of MATLAB.

In the first phase, the parameter (*k*) of *k*-means is not expected to significantly impact the expected result. To verify this, Figure 3.10 shows the clustering results with different values of *k*.

As the proposed algorithm needs to preserve only the top segments to obtain spikes, all of the results in Figure 3.10 can be used; however, if a small *k* value is selected, a portion of stems is considered as part of the top segments; this introduces an error in spike

height. If a bigger value of *k* is selected, the stem points counted may be less. However, a perfect parameter value that can completely remove all of the stem's points cannot be guaranteed. Furthermore, as the value of *k* increases, the efficiency of the algorithm may reduce. This is discussed later.

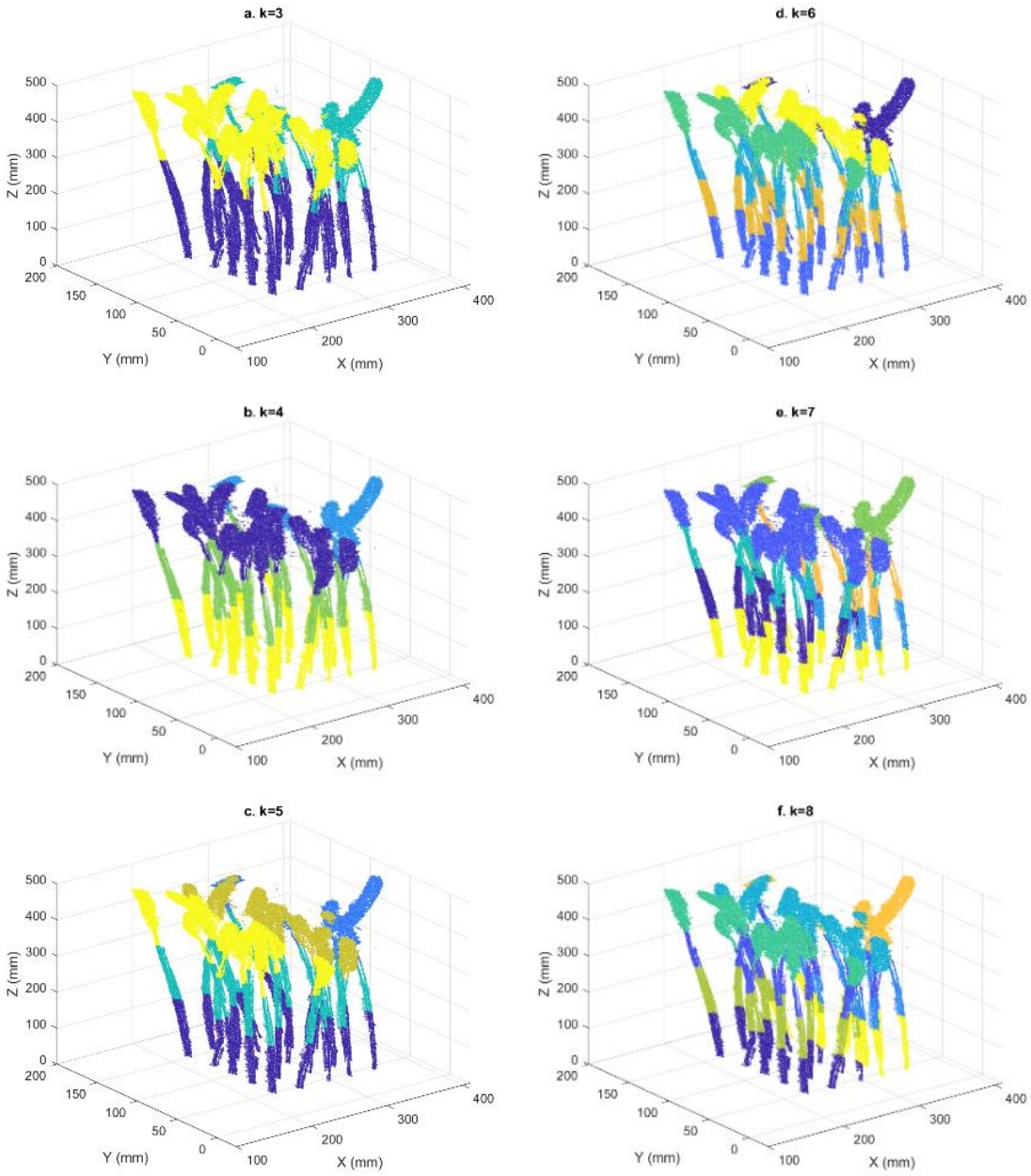


Figure 3. 10 – Clustering results with different values of *k*. As the value of *k* increases, the number of clusters increases incrementally. However, using Algorithm 3.1, the top clusters belonging to the spikes can be obtained regardless of the *k*.

In the second phase (Algorithm 3.2), the 3D point cloud image is projected onto the 2D point cloud (top view), thus reducing the point distance in the within-cluster; this is

important because the height of the spike is longer than the width and length in a 3D space. Further, it improves the ability of the algorithm to identify individual spikes. To validate the performance of this phase, different scenes were tested with the proposed algorithm. As shown in Figure 3.11, in these three scenes, some wheat crops were dense or mutually overlapping (highlight areas). However, evidently, from the clustering results (Figure 3.12), the proposed method still exhibited superior robustness and feasibility compared with traditional algorithm results.

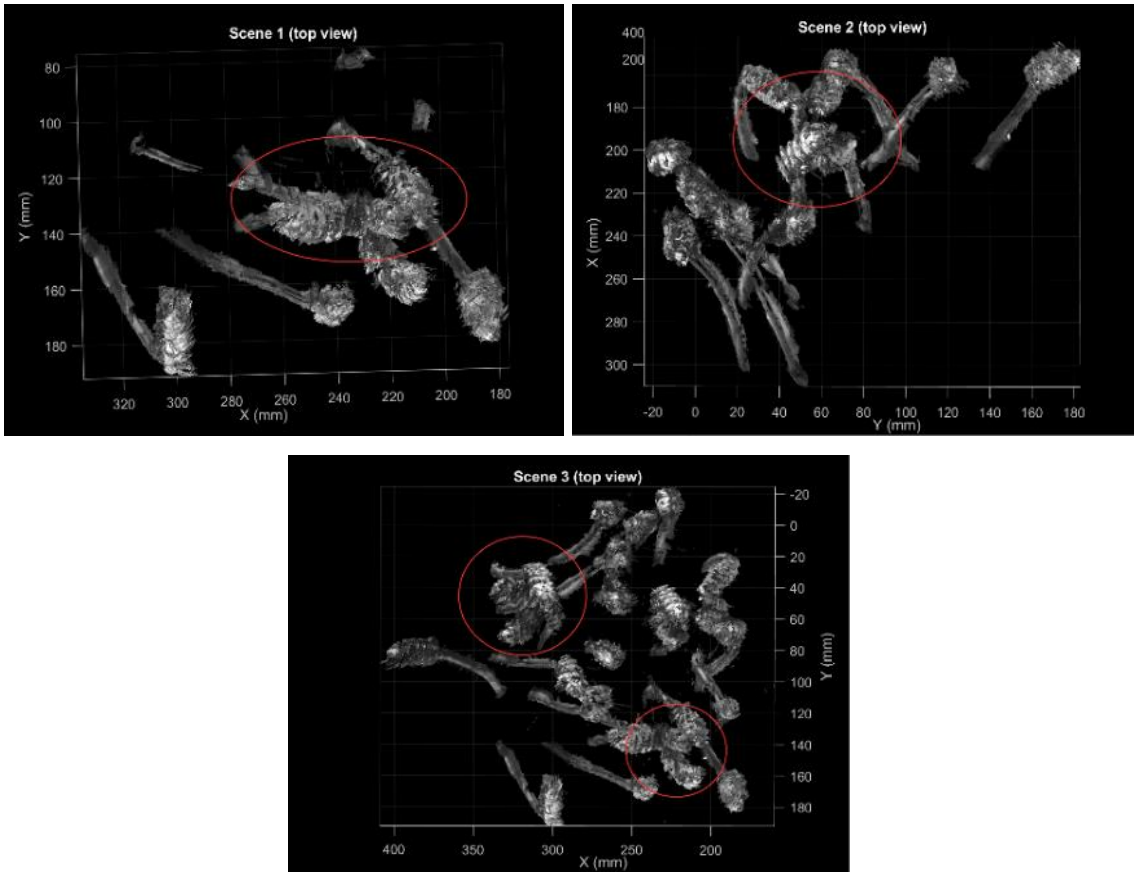


Figure 3. 11 - Three different scenes where the wheat crops are dense (particularly in the highlighted area).

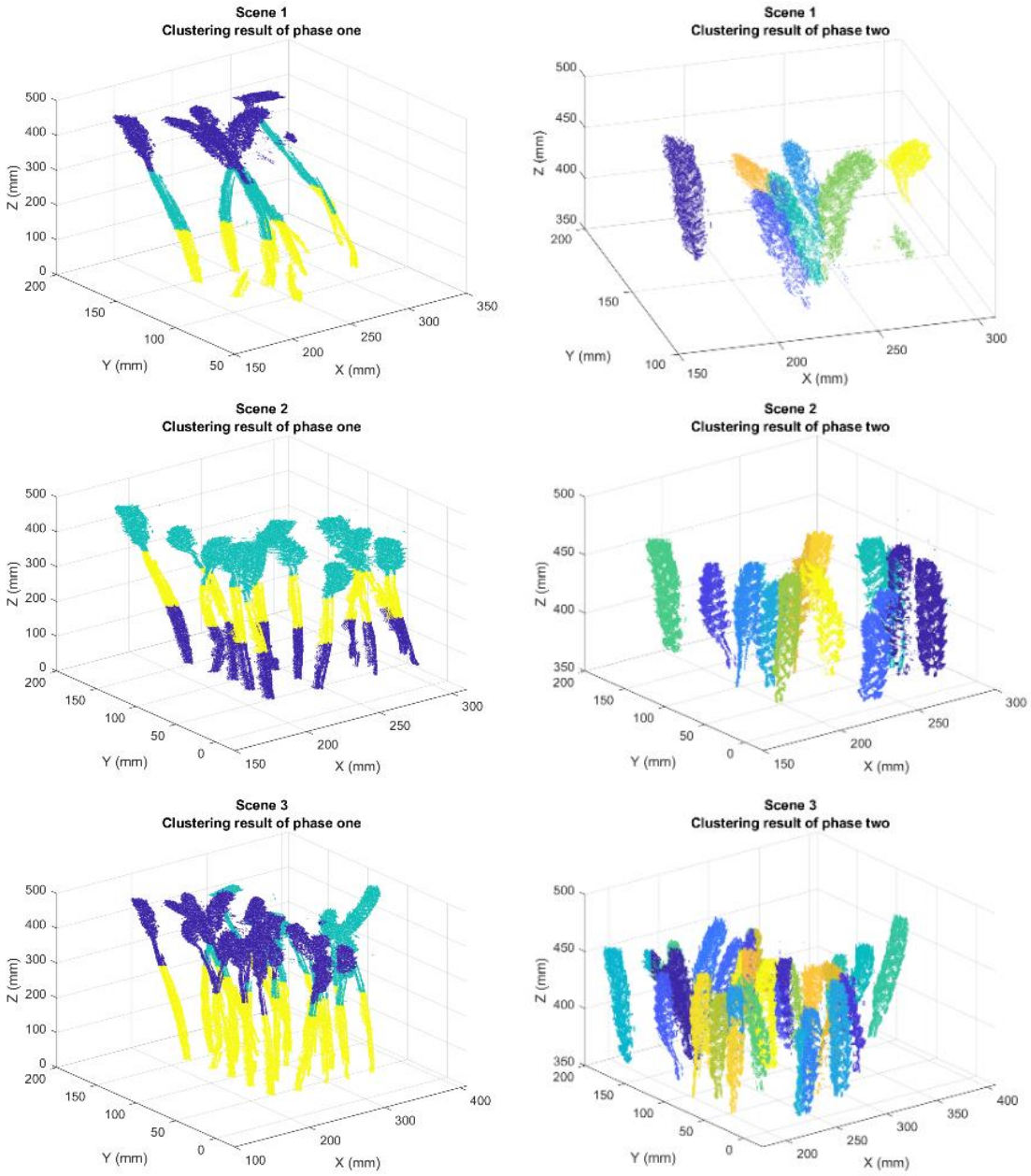


Figure 3. 12 – Clustering results with different scenes based on the proposed algorithm. From the first to the second phase, the wheat spikes are separated from the wheat plants, subsequently, each spike is identified.

3.4.2 Efficiency analysis of the proposed algorithm

To analyse the efficiency of the algorithm, the same laptop as mentioned previously was used to run the algorithms for different situations of the wheat crops. The algorithm was run five times for each of the three situations in the above figure, and the average time was recorded. The value of *k* was set as six for phase one and the max iterations of

the RANSAC algorithm (in phase two) as 1,000. The average running times of phase one and phase two are recorded in Table 3.2.

Table 3. 2 – Average running time of the proposed algorithm

Scene number	Number of points	Running time for phase one	Running time for phase two
1	166283	3.26 s	44.27 s
2	747982	8.84 s	184.13 s
3	902813	12.51 s	377.68 s

As illustrated in Table 3.2, owing to the performance of Lite *k*-means, the *k*-means in the proposed algorithm was not computationally taxing. A comparison of Tables 3.1 and 3.2 indicates that the different parameter values that influence the calculation time were evident but the changes were insignificant (in Table 3.1, *k* is three). In phase two, the algorithm operates with self-adaptive updating of the parameters and calls RANSAC to fit the shape of each cuboid; this part is crucial in the efficiency of the algorithm. Throughout the entire process, the efficiency of the algorithm in processing wheat was excellent. However, the running time for handling a field image with one square meter increased considerably. Essentially, because of the volume calculation, the calculation time was spent primarily in dividing all spikes into 3,000 segments, and RANSAC was called to fit each segment to evaluate the total volume. Therefore, the parameter *k* was set as 3,000 to conduct segmentation and then realise shape fitting. Thus, the method called the RANSAC 3,000 times to fit the shape of each small segment for the volume calculation of spikes; however, the proposed method required approximately 30–40 min to perform volume calculation on a standard modern laptop.

3.4.3 3D field capture

To analyse the field trials in detail, first, the 3D point field capture system was introduced. A portable and field-deployable solution that can completely image a field-grown trial plot of dimensions 2 m × 5 m × 1 m in less than 1 s was constructed for

analysis with wheat identification algorithms. The platform was adjustable to accommodate typical weather conditions, including direct solar illumination, and be self-powered. The solution deployed in the fields during 2020 is shown in Figure 3.13; it included four structured light scanners from Photoneo s.r.o., each positioned parallel to one side of the trial plot edge and orientated at 45° to the vertical. The arrangement enabled capturing the central region of the plot, neglecting only 300 mm around the edges, which are normally excluded from analysis in most trials. Each scanner was triggered in sequence to avoid interference and a region of 2 m × 2 m × 1.5 m was captured in approximately 5 s. The scanners were optimised to overcome bright ambient light using structural netting above and to the sides of the mounting frame. Additionally, the selection of the scanner's exposure, laser brightness, and processing algorithms was critical.

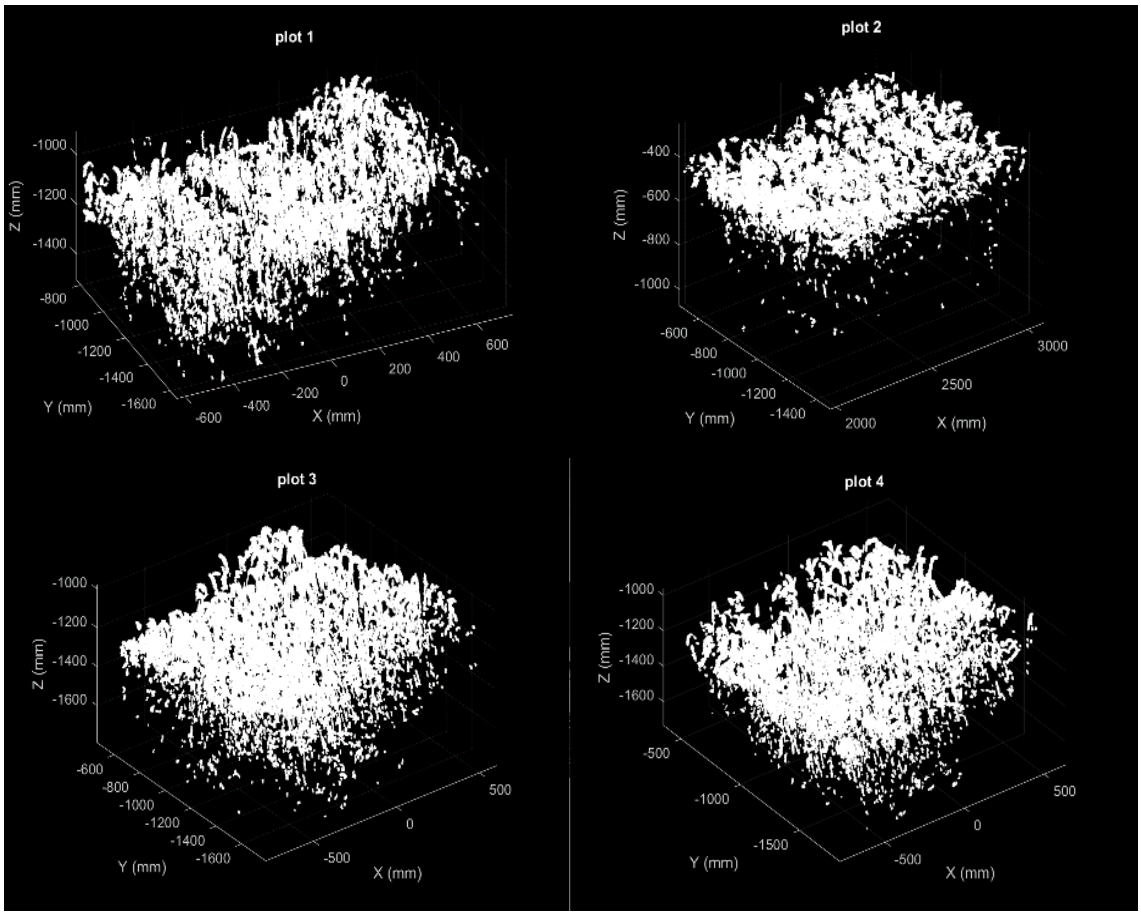
Figure 3. 13 – Field use of 3D capture system incorporating four scanners.

The four independent scans were reconstructed into a single point cloud using a common reference chart placed in all scanners, as shown in Figure 3.13. This chart did rendered the auto-alignment function of the scanners redundant, but as chart reflectivity issues already produced variable results outdoors no functionality was lost in reality. Instead, an in-house algorithm was created to locate the reference chart within each

scanner point cloud and produce a translation matrix to align all four into a single virtual replica of the trial plot. Unlike single-point measurement systems, the final point clouds include information on the complete surface for all the wheat heads, detailed to the grain level. The final point clouds were cleaned for noise using a statistical outlier filter and the resolution was reduced using a sub-sampling algorithm to reduce the computational power needed for the next stage of processing, identifying spikes, and performing dimensional measurements.

3.4.4 Comparison of manual measurement with the proposed method

To verify the performance of the proposed method for field application, five different field plots captured by the platform were selected and cropped for testing. Each plot was approximately 1 m^2 of a wheat field, and the original images used are shown in Figure 3.14.



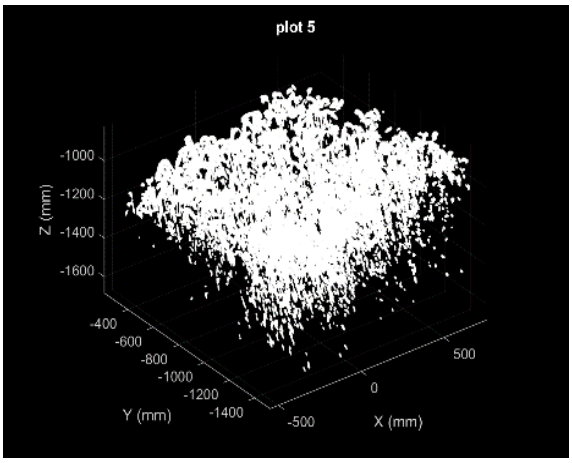


Figure 3. 14 – Five different 3D point images from the field. Each image contains approximately 200 wheat plants.

For manual measurement, a sample area is typically selected in the field. The number and size of spikes in the sample area are measured to infer the total number and average size of wheat crops in the entire field. In this experiment, for each scenario, a square of 0.25 m^2 was selected as the sample area. The number of spikes was counted and the average size of spikes (height h_m and width w_m) in the sample area was measured. The amount of wheat (spikes m^{-2}) was calculated according to the following equation.

$$num_m = \frac{n_m}{0.25} \tag{3.1}$$

The proposed method was used to calculate the total volume V_a of all spikes and the average size of a single spike. Algorithm 3.2 was used to compute the height, length, and width (h_a, l_a, w_a) of the spike, and cuboid fitting was used to facilitate comparison with manual measurement. The values h_a and $w'_a = (l_a + w_a)/2$ were compared with h_m and w_m, respectively. As each tested scenario was approximately 1 m^2 of a wheat field, for the proposed method, the total volume of spikes was divided by the volume of a single spike to estimate the number of spikes in each scenario, as follows.

$$num_a = \frac{V_a}{h_a \times w'_a \times w'_a} \tag{3.2}$$

The comparison results are recorded in Table 3.3. To compare the proposed method with the manual method, Eqs. 3.3–3.5 were used to estimate the error rate of each plot.

Table 3. 3 – Comparison of results between manual measurement and the proposed method

Plot number	Average size		Number of spikes		Total volume
	Manual	Proposed method	Manual	Proposed method	
	h_m / w_m	h_a / w'_a	num_m	num_a	V_a
1	83.4/13.5 mm	76.7/12.4 mm	212	173	2042491 mm³
2	71.9/15.5 mm	63.6/14.6 mm	260	202	2766830 mm³
3	84.2/14.2 mm	81.8/18.2 mm	200	259	7020030 mm³
4	82.4/15.2 mm	81.3/17.6 mm	212	207	5179640 mm³
5	78.3/15.0 mm	76.4/15.6 mm	228	208	3866860 mm³
Standard deviation	5.1/0.8 mm	7.4/2.3 mm	/	/	/

Error rate in the number of spikes: $$Error_1 = \frac{|num_m - num_a|}{num_m} \quad (3.3)$$

Error rate in the spike height: $$Error_2 = \frac{|h_m - h_a|}{h_m} \quad (3.4)$$

Error rate in the spike width: $$Error_3 = \frac{|w_m - w_a|}{w_m} \quad (3.5)$$

Table 3. 4 – Error rates of the proposed method

Plot Number	$Error_1$	$Error_2$	$Error_3$
1	18.40%	8.03%	8.15%
2	22.31%	11.54%	5.81%
3	29.5%	2.85%	28.17%
4	2.36%	1.34%	15.79%
5	8.77%	2.43%	4%
Average	16.27%	5.24%	12.38%

As illustrated in Table 3.4, the three average error rates defined for the five experiments above were 16.27 %, 5.24 %, and 12.38 %.

3.5 Summary

A high-throughput field capture platform for wheat combined with an unsupervised automatic measurement of wheat spikes based on an adaptive *k*-means algorithm with dynamic perspectives was proposed. This platform can handle complex environments where hundreds of wheat spikes are grown densely. This method provides a novel framework to obtain wheat spike dimensions and total volume, instead of manual measurement.

These experiments helped perform a detailed analysis of the proposed *k*-means algorithm. Although *k*-means is an uncertain algorithm, which cannot ensure reliable outputs, for the proposed algorithm, the clustering result was sufficient for shape fitting. Additionally, the shape fitting algorithm was not the focus of this work; nonetheless, the cuboid fitting results for straight spikes were superior to those of curved ones. This is because cuboids cannot accurately fit the height of curved wheat spikes. As presented in Table 3.3, all of the average heights obtained by the proposed algorithm were slightly smaller than those measured manually. This is because as most of the tested wheat spikes were slightly curved, using cuboid shape fitting resulted in some errors. In addition to the shape fitting algorithm, owing to the spikes being more curved, the overlapping in the top view was distinct. This might influence the clustering result of the proposed *k*-means algorithm.

Furthermore, the proposed method presents some issues, which can be addressed in future work. First, a self-adaptive *k*-means algorithm to update the *k* iteratively in Algorithm 3.2 was proposed for spikes counting. However, for volume calculation, the computational efficiency was poor. Therefore, when handling field images, the entire image was divided into three segments. Second, as mentioned above, the accuracy of this method was affected by the curvature of the spike. Five field data set results were used

and the average error in the number of spikes exceeded 16 %, of which two errors exceeded 20 %. The performance of the algorithm may decrease if this analysis is extended to spike dimensions assessment for other field datasets.

Overall, the experiment results imply that the proposed method can be developed as a tool to evaluate the size and yield of wheat spikes, particularly for straight spikes, thus avoiding laborious manual measurements. As method performance can still be improved to handle curved wheat spikes, future work can further optimise the algorithm to handle the environment where the wheat spike is arched. In addition to wheat, the method can be extended to barley, corn, and fruit phenotyping applications.

Chapter 4

Configurable Crop Perception II: Identification of Soft Fruit

In recent years, object recognition based on deep learning has gained prominence due to its wide range of applications. More recent developments in deep learning-based object detectors have been detailed and surveyed in [89]. However, its learning process relies on large amounts of labelled data and powerful computational resources. For agricultural robots, recognition is the first step, and adopting appropriate harvesting strategies based on recognition results is crucial. This chapter focuses on the corresponding models and algorithms involved in perception systems, which were integrated with the action system herein to guide the robot in harvesting.

4.1 Soft Fruit Recognition Based on Conditional Generative Adversarial Networks

Although crops, such as corn and wheat, can be harvested in bulk, soft fruits in the greenhouse, such as strawberries, still require manual picking. Therefore, to enable the harvesting robot for this task, mature strawberries must first be recognised and localised. In this section, a perception system based on conditional generative adversarial networks (cGANs) was proposed to identify ripe fruits using synthetic data. To apply the system to a strawberry greenhouse, a new factor, defined as the clustering complexity of strawberries, was defined. More details are presented below.

4.1.1 Synthetic dataset

Although deep learning has played a pivotal role in the target recognition, data collection and labelling are time-consuming, particularly when dealing with complex environments and light conditions. Further, travel restrictions are continuously changing owing to COVID-19; thus, collecting a large amount of data from the field is more tedious. Synthetic datasets have been effectively used in research; thus, creating and using synthetic datasets can address these concerns [90], [91]. This study generated a synthetic dataset by combining fruit and background images. In particular, various background images were gathered from the Internet, and pictures of the farm were captured. The pictures with the most similar backgrounds and colours (green/brown) as those of the field (see Figure 4.1 a) were selected. Subsequently, crops were placed on top of the background. Herein, pictures of individual strawberries from a fruits dataset, namely, Fruits-360 dataset [92], were placed on the background image to synthesise data. Each strawberry was captured from its white background, and lightning variation was accomplished using gamma correction, a common nonlinear operation for image illumination. Additionally, to create irregular crop images, a bitwise-and operation was applied to the target crop image and binary mask. Eleven masks were used in this dataset, and they consisted of random lines and blobs emulating obstacles present in natural environments. The constructed dataset fit this objective using strawberries from Fruit-360. Finally, the synthetic dataset contained 4,500 instances, with 900 instances for each fruit. The process is shown in Figure 4.1 b. According to the synthetic process, the input image and ground truth for model training can be obtained simultaneously (Figure 4.1 c).

Therefore, the advantage of this method is that the dataset required for training is automatically generated, with high efficiency and no manual labelling. Existing popular object detection models require customising their datasets and labelling is time-

consuming. Moreover, current popular labelling processing software, such as [93], requires handling each image in front of the screen. Even if a picture takes a few seconds, the overall working time of thousands of pictures becomes considerably large and cannot be ignored. Whereas in the proposed method, all training pictures can be automatically generated within a few minutes, and network training can begin immediately.

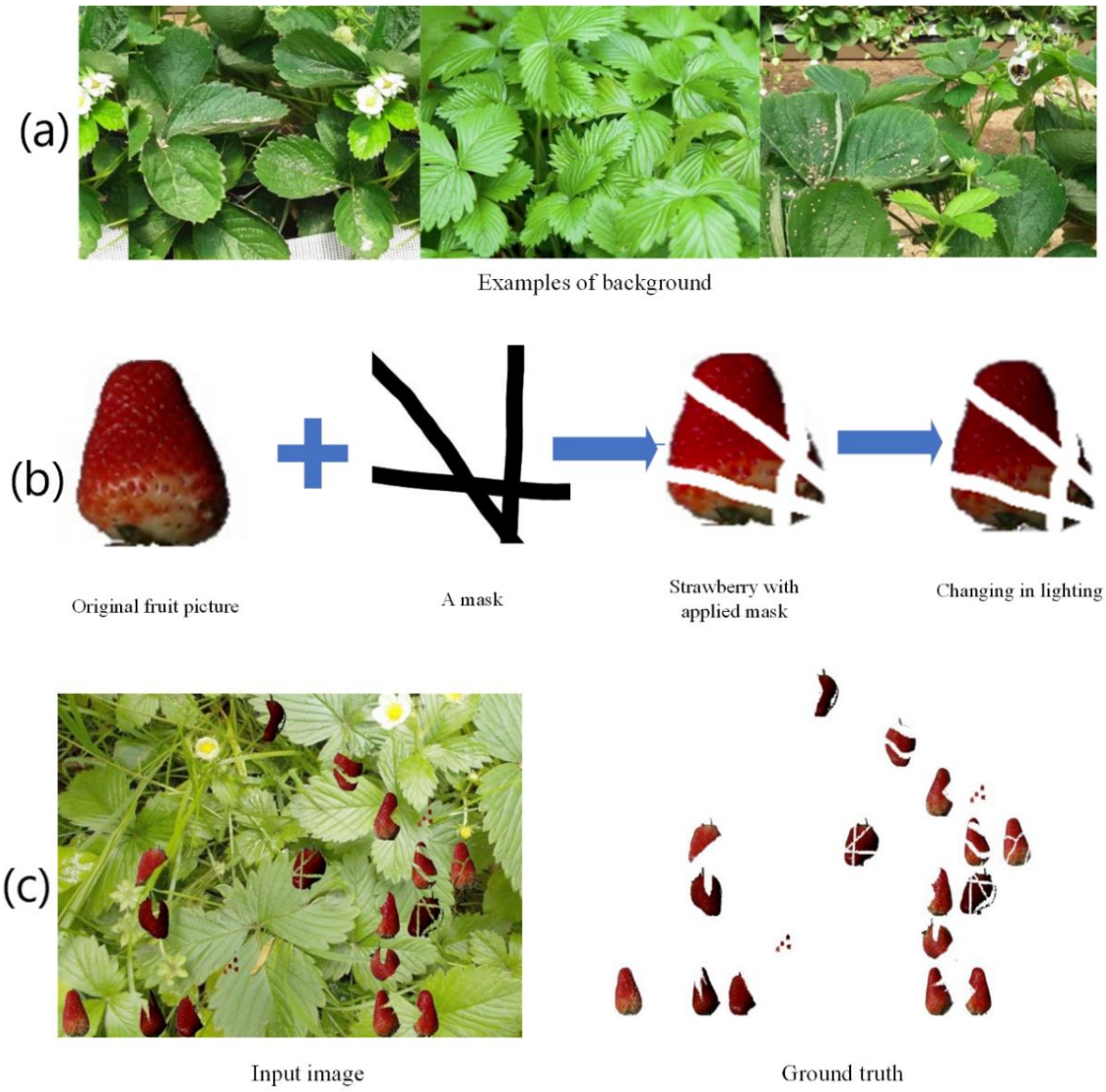


Figure 4. 1 - Generation process of the training dataset. (a) Some examples of background; (b) Obtaining the final image from the original fruit picture by applying masking and lighting changes; (c) Sample of the synthetic dataset.

4.1.2 The perception system based on the GAN

To use synthetic dataset for training the perception system model, the pix2pix model, a cGAN, was introduced [94]. The pix2pix model was designed to perform image-to-image translation; thus, it can translate an input image into a corresponding output image using the generator of a cGAN. As described in the previous section, the synthetic data generates a pair of images simultaneously, such as the left and right images in Figure 4.1 c. The core of the pix2pix model chosen for this work is to use the model's picture translation function to map the complex image (the left one in Figure 4.1 c) to the simple image (the right one in Figure 4.1 c).

The core idea is to ensure the generator has a good mapping capability by training the cGAN. Thus, the generator can map the input image to a ground truth thereby fooling the discriminator.

Generally, cGANs learn a mapping from an observed image x and random noise vector z to y, $G:\{x,z\}\rightarrow y$. The generator G is trained to produce outputs that cannot be distinguished from "real" images by an adversarially trained discriminator, D, which is trained to do as well as possible at detecting the generator's "fakes". The training procedure is illustrated in Figure 4.2.

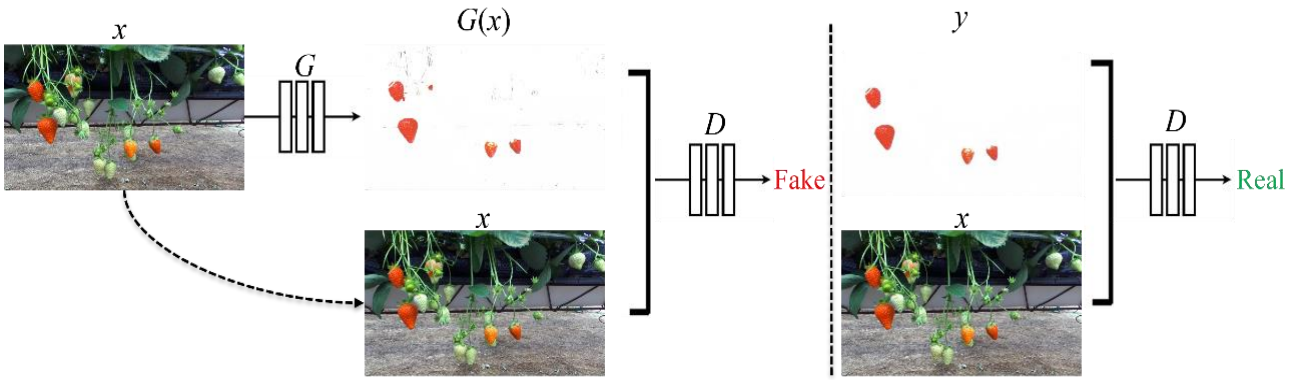


Figure 4. 2 - Training a cGAN to map a real farm picture → the picture only contains ripe strawberries. The discriminator learns distinguishing between fake (synthesised by the generator, G(x)) and real tuples (ground truth, y). Both the generator and discriminator observe the input image x.

The value function of a cGAN can be expressed as follows.

$$L_{GAN}(G,D) = \mathrm{E}_{x,y}\left[\log D(x,y)\right] + \mathrm{E}_{x,z}\left[\log\left(1 - D\left(x, G\left(x,z\right)\right)\right)\right], \tag{4.1}$$

where *G* attempts to minimise this value function against an adversarial *D*, which attempts to maximise it. In the pix2pix, the discriminator's job remains unchanged; however, the generator is tasked with fooling the discriminator and being near the ground truth output in a Manhattan distance (L1) sense. This is because previous approaches have reported that mixing the GAN objective with a more traditional loss is beneficial [95].

$$L_{L1}(G) = \mathrm{E}_{x,y,z}\left[\left\|y - G(x,z)\right\|_1\right]. \tag{4.2}$$

Therefore the final objective of pix2pix is as follows.

$$\arg\min_{G}\max_{D} L_{GAN}(G,D) + \lambda L_{L1}(G). \tag{4.3}$$

According to the analysis work of the objective function [74], the cGAN alone (setting $\lambda = 0$ in Eqn. 4.3) achieves sharper results but introduces visual artefacts on certain applications. Adding both terms together (with $\lambda = 100$) reduces these artefacts.

According to our synthetic data, this map/translation idea was introduced into fruit detection; for example, in Figure 4.1 (c), the pix2pix model can map the left image into the right image. Because only mature strawberries need to be determined, this pix2pix model can make detecting crops in a complex environment simpler, regardless of the complexity of the background environment. The original model worked with 256 × 256 images, and as the dimensions of the images increase, the model quality decreases. Thus, an improved model called Pix2pixHD [96] was introduced into our perception system to handle bigger images. Pix2pixHD is an improved pix2pix framework and uses a coarse-to-fine generator, multi-scale discriminator architecture, and robust adversarial learning objective function.

Pix2pixHD decomposes the generator into two sub-networks: a global generator and local enhancer networks. The global generator network operates at a resolution of 1024 × 512, whereas the local enhancer network outputs an image with a resolution that is 4 × the output size of the previous one. To differentiate high-resolution real and synthesised images, Pix2pixHD uses three discriminators (D_1, D_2, D_3) with an identical network structure but operate at different image scales. The real and synthesised high-resolution images are down-sampled by a factor of 2 and 4 to create an image pyramid of 3 scales. Finally, to improve the GAN loss in Eq. 4.1, a feature-matching loss based on the discriminator was incorporated. The feature matching loss function is expressed as follows:

$$L_{FM}\left(G, D_K\right) = \mathrm{E}_{(x,y)} \sum_{i=1}^{T} \frac{1}{N_i} \left[\left\| D_k^{(i)}\left(x, y\right) - D_k^{(i)}\left(x, G(x, z)\right) \right\|_1 \right], \tag{4.4}$$

where $D_k^{(i)}$ represents the feature of the *i*-th layer extracted by the discriminator D_k ; T is the total number of layers; and N_i denotes the number of elements in each layer.

The full Pix2pixHD objective combines both GAN and feature-matching losses, as follows:

$$\min_G \left(\left(\max_{D_1, D_2, D_3} \sum_{k=1,2,3} L_{GAN}\left(G, D_k\right) \right) + \lambda \sum_{k=1,2,3} L_{FM}\left(G, D_k\right) \right). \tag{4.5}$$

The feature matching loss L_{FM} serves only as a feature extractor and does not maximise the feature matching loss.

After the map/translation work, the watershed algorithm [77] was used to estimate and divide the number of strawberries. The watershed segmentation algorithm is applied to an image's gradient rather than the image itself. It is based on the concept that regions are characterised by small variations in grey levels and have diminished gradient values. In formulating watershed segmentation, the regional minima of catchment basins

correlate positively with the diminished value of the gradient corresponding to the objects of interest. Figure 4.3 shows that image (a) is output from the pix2pix and contains only ripe strawberries, which is relatively easy for the watershed segmentation algorithm to analyse the image's gradient and segment strawberries. Subsequently, the segment information (bounding box) can be applied to the original image (Figure 4.3 b→c).

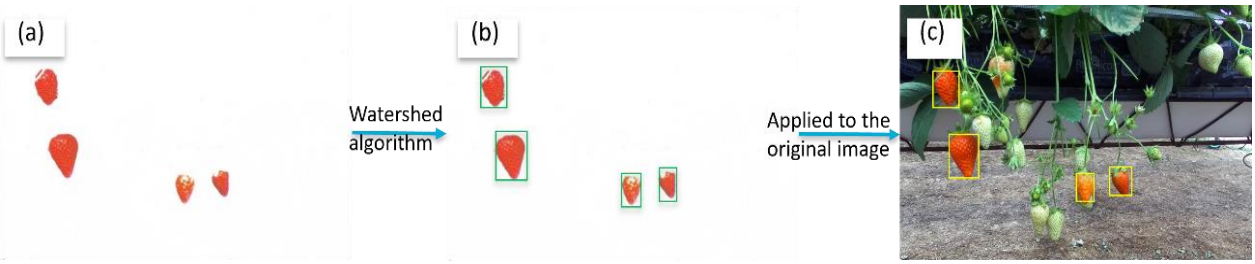


Figure 4. 3 - Watershed segmentation to circle each detected strawberry. (a) GAN maps the original image into a picture; (b) Subsequently, it applies the watershed algorithm to obtain the bounding boxes; (c) Finally, the bounding boxes are placed on the original image to obtain the final result.

To illustrate the entire process of strawberry detection and segmentation, the overall architecture of the proposed perception system is illustrated in Figure 4.4. Figure 4.4 shows that the Pix2pixHD model receives the 2D image from the stereo camera and inputs the translated image into the watershed algorithm for crop detection. Subsequently, the camera combines the 2D information and accesses the 3D point cloud to localise the crops.

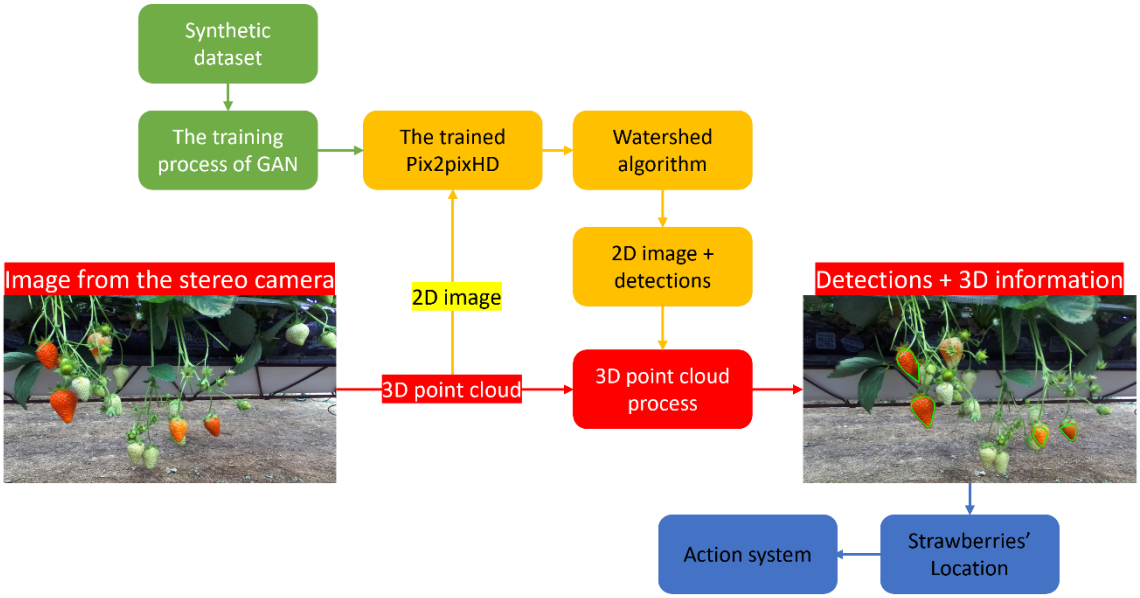


Figure 4. 4 - Overall architecture of the perception system based on the GAN. The green arrows indicate image training; yellow arrows indicate the target detection process; red arrows indicate the acquisition of 3D information of the target; blue arrows show the activation of the action system.

Additionally, the proposed system can be extended to harvest other crops by changing the synthetic dataset and end-effector. As shown in Figure 4.5, the strawberry dataset was switched to tomato for training a new perception model. This study focused on the strawberry application; more details regarding the performance of the model are discussed later.

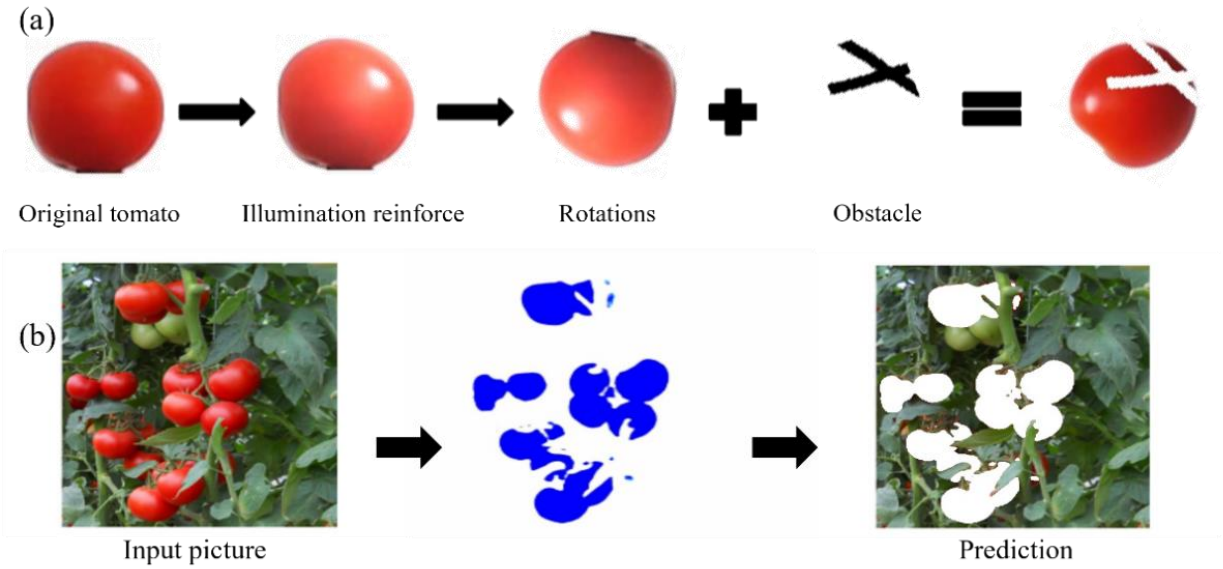


Figure 4. 5 - (a) During data synthesis, the strawberries are replaced with tomatoes; the diversity of training data is increased by randomly changing the illumination and rotation. (b) Example of tomato model predictions.

4.1.3 Evaluation of experimental results from field trials

The proposed perception system comprises both identification (detect the maturated strawberries) and localisation (see [97] for more details of the point cloud generated by the camera). To test the perception system's validity and performance, actual images were collected from a strawberry greenhouse to evaluate the proposed perception system. First, several images containing different conditions and multiple strawberries were selected to

test the model (six example images are shown in Figure 4.6). The images in the left column were captured on a sunny day, whereas the images on the right were captured on a cloudy day. The results revealed that the proposed perception system can effectively detect ripe strawberries.

Figure 4. 6 - Strawberry detection and localisation in natural conditions.

After the detection, the detected regions were cropped from the original image (Figure 4.7(a)) and the remaining undetected sections or complete ripe strawberries were analysed (Figure 4.7(b)). If a strawberry was partially detected, then the undetected section was not considered (Figure 4.7(c)). This is because the robot is expected to explore that area using the information on the detected portion and better detect the entire target. This method can easily detect any crops that are visually undetected by the system.

Using this testing condition and measurements, ripe strawberries in the images selected can be detected. However, the system presents 81.4 blobs per image, and each image contains at most 30 visible strawberries.

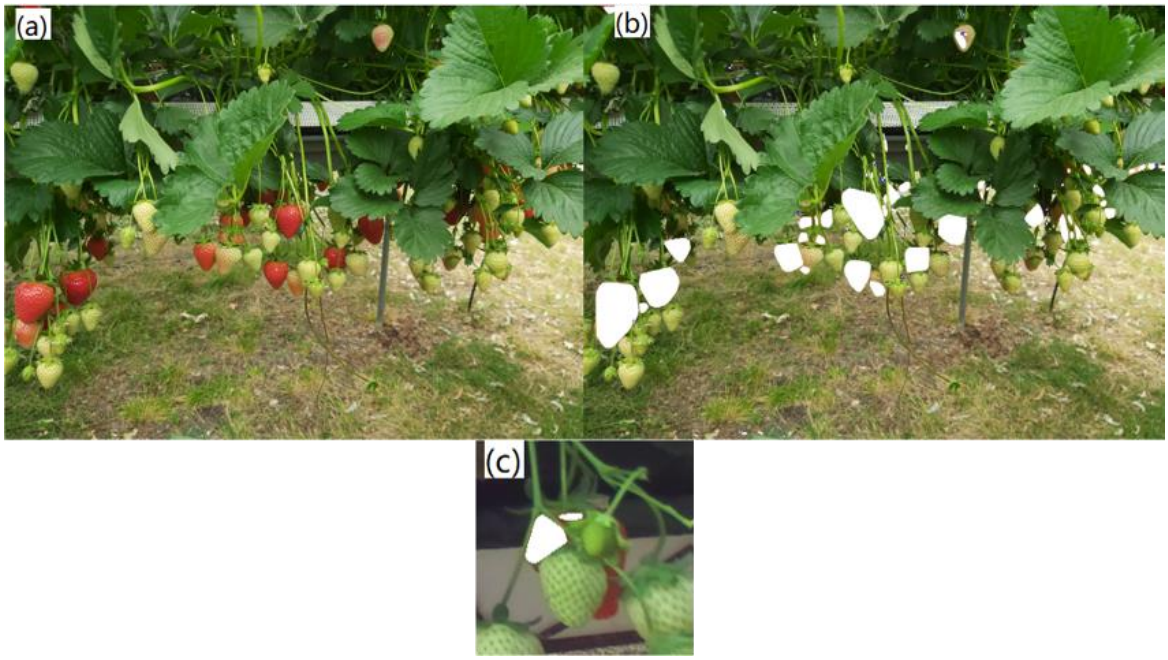


Figure 4. 7 - Performance measurement example. (a) Original image; (b) Remaining undetected sections after recognition; (c) Partially detected strawberry.

Further, small blobs (noise) can be eliminated and extremely close strawberries in the perception system can be segmented, each using the following operations. The former requires a morphological operation [98] that eliminates noise. Furthermore, the watershed algorithm allows for counting the objects or further analysis of the separated objects (see [99] for the algorithm implementations in open-source libraries). The application comparison results of these two operations are illustrated in Figures 4.8 and 4.9, respectively. Although the two operations can improve the performance of the perception system, they cannot ensure that all ripe strawberries are accurately divided. To analyse this performance in particular, 50 images were captured from the farm to estimate the error rate in the number of strawberries. First, the perception system was used to detect and count ripe strawberries in each image; next, this data was compared with that of

manual counting. The following equation was used to estimate the error rate in the number of strawberries.

$$Error = \frac{\left| num_m - num_p \right|}{num_m}, \tag{4.6}$$

where, num_m denotes the number of ripe strawberries counted manually; and num_p denotes the perception system output. For all fifty testing images, Eq. (4.6) was used to estimate the error rate of each image, and the average error rate was calculated as 10.83 %.

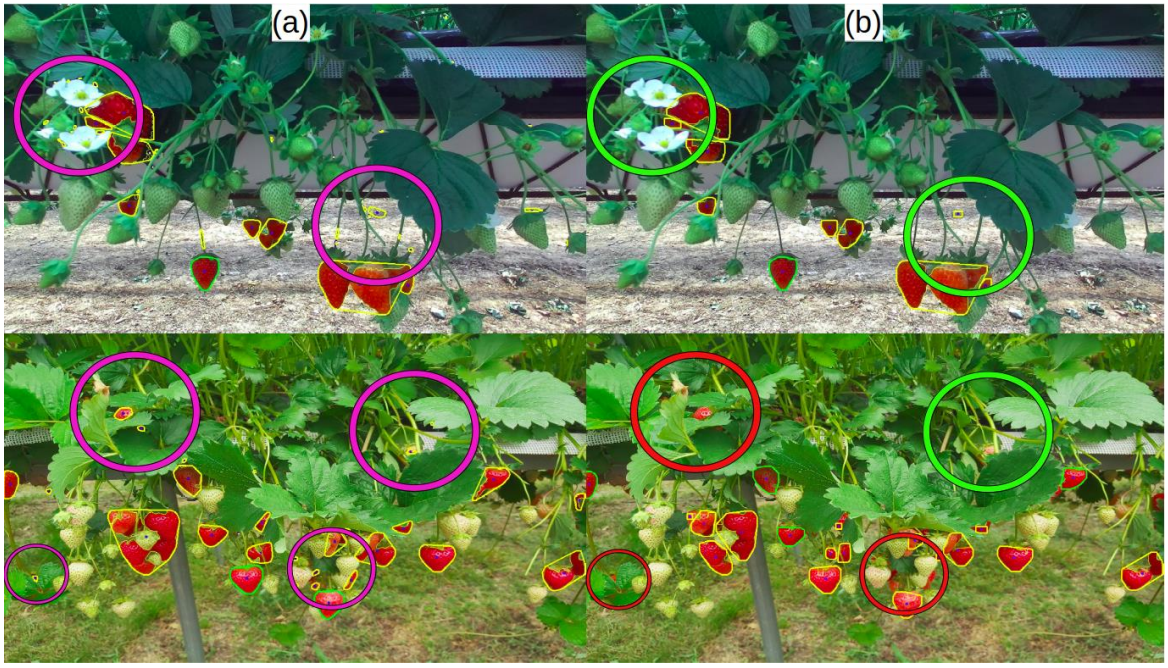


Figure 4. 8 - Morphological operations: (a) predictions without operations applied, and (b) predictions with operations applied. Purple circles indicate areas with small blobs, green circles represent areas where noise is eliminated, and red circles represent some correctly localised crops.

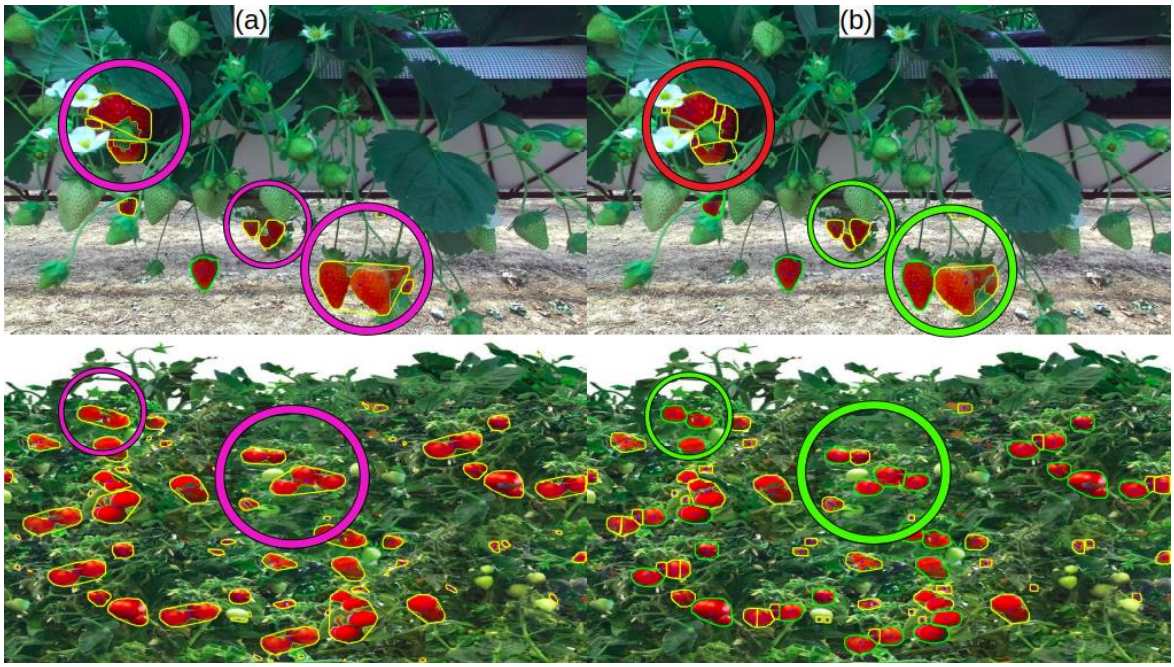


Figure 4. 9 - Applied watershed algorithm to blobs with an area larger than 3,000 pixels. (a) Predictions without the watershed algorithm; (b) Predictions with the watershed algorithm. Purple circles indicate blobs that the watershed method will be applied; Green circles indicate where the blobs cluster was correctly divided, and red ones when they were not.

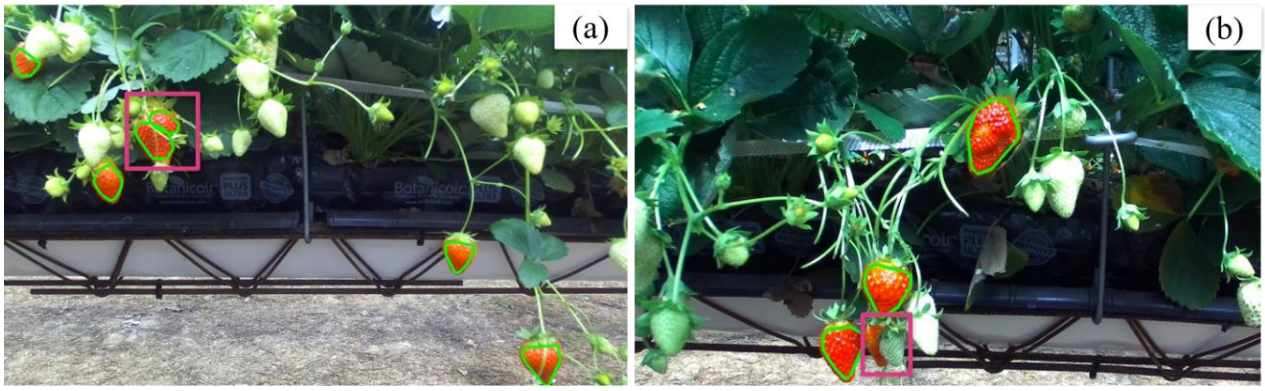


Figure 4. 10 - Situations where the perception system cannot accurately count strawberries. (a) Wrong segmentation; (b) Overlapping.

Figure 4.10 shows situations where the perception system cannot accurately count all strawberries. The error rate was primarily owing to the occlusion. Occasionally, a single strawberry was divided into two because of stems (see highlight area Figure 4.10 (a)). Further, the perception system could not always recognise the overlapped strawberries (Figure 4.10 (b)).

4.2 YOLACT-based Fruit Cluster Complexity Analysis

As in real-world environments, some ripe strawberries are surrounded by stems and unripe strawberries; harvesting such target strawberries is challenging for robots. Although the above perception system can recognise target strawberries, it cannot determine the ease of harvesting them. The robot can benefit from selective picking if it can determine the complexity of picking the strawberry. For example, if the hardware and software of the robot are insufficient to support its selection of high-complexity strawberries, it can ignore this category of strawberries to reduce the picking damage rate.

Recall that strawberry cluster complexity reflects the difficulty in harvesting when the target strawberry is surrounded by other strawberries and leaves. The more the obstacles, the more the difficulty for the manipulator to pick smoothly without damaging to the strawberry.

To allow the robot to handle strawberries with different harvesting complexities in a more targeted manner, this section presents another deep-learning model that was used to classify the cluster complexity level. First, to describe this situation, this study categorised complexity into three categories: easy (no occlusion), medium (little occlusion), and hard (significant occlusion).

Figure 4. 11 - Strawberry clusters with different complexity levels. The more densely distributed and heavily overlapped the strawberries, the higher the cluster complexity.

As shown in Figure 4.11, when a strawberry is occluded by obstacles, the difficulty in picking it depends on the degree of occlusion presented by its obstacles. When a strawberry is surrounded by obstacles, part of its pulp is covered; this can be used as a key feature to learn the different levels of complexity. Moreover, this feature can be reflected in instance segmentation. To utilise this, each strawberry must be labelled using image polygonal annotation [100], as shown in Figure 4.12. If the strawberry is classified as hard to harvest, it may contain a few small segments. With this annotation method, the generated labels contain categories (easy, medium, hard) and vertex positions of each polygon, then the labels can be used for training. Therefore, “easy, medium, hard” presents the complexity, and vertex positions of the polygon present the segmentation information.

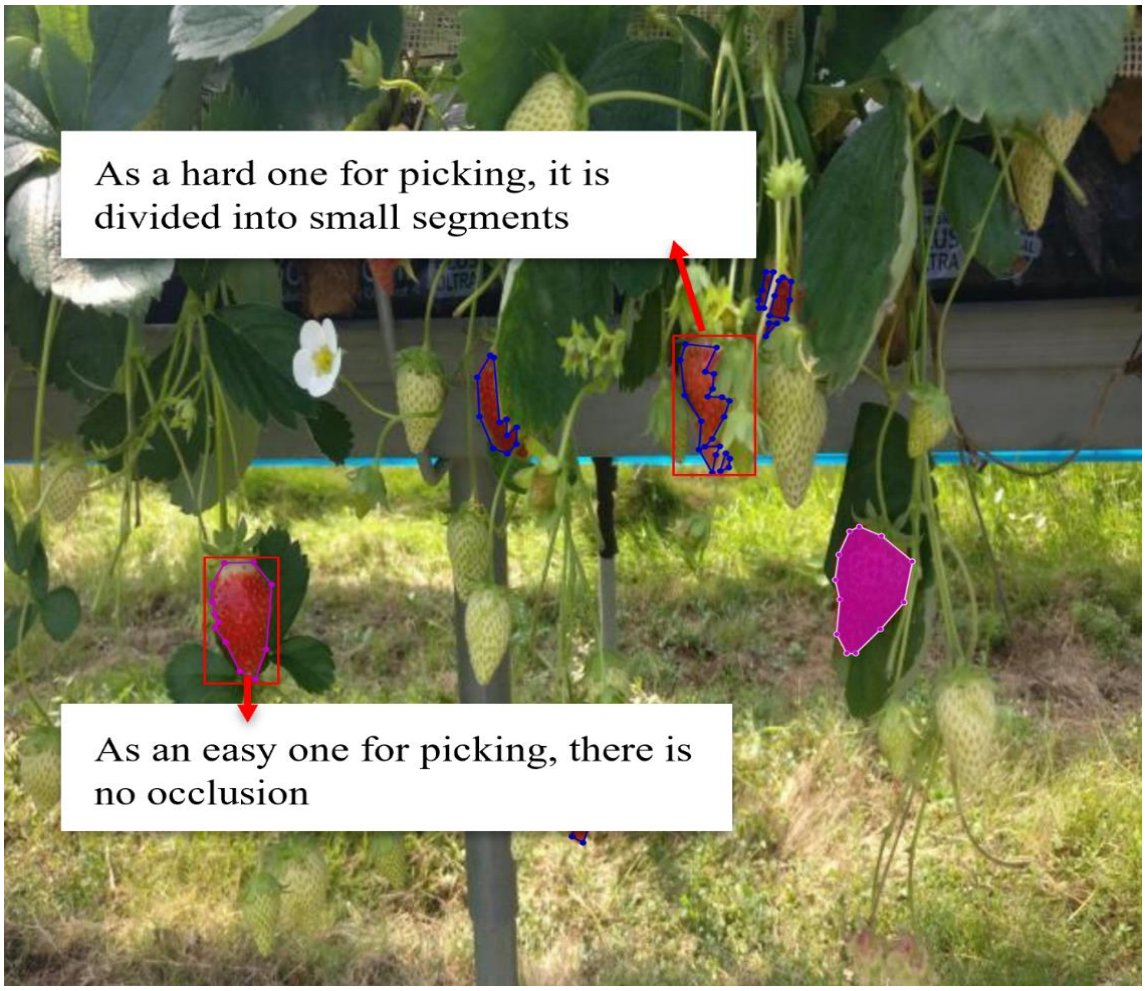


Figure 4. 12 - Example of image polygonal annotation. During labelling, strawberries that are not covered are labelled as easy for picking. If half or more of the body of the strawberry is covered, it is labelled as hard for picking; otherwise, it is labelled as medium for picking.

4.2.1 YOLACT instance segmentation method

YOLACT [101] was selected to perform the classification function as it provides real-time instance segmentation. It predicts mask prototypes and per-instance mask coefficients in parallel, and linearly combines them to form the final instance masks. The YOLACT architecture is based on RetinaNet [102] using ResNet-101 + FPN. The first branch is the prototype generation branch, which is a semantic segmentation model, and is implemented based on FCN [103]. Whereas the last layer has k channels corresponding to k prototype masks. The second branch adds a prediction head network to the object detection branch to generate $(4+c+k)$ predictions, i.e., k mask coefficients of each

anchor, four coordinates, and *c* category confidence of the bounding box predicted by the object detector for each anchor. Finally, to produce instance masks, the results of the prototype and predicted mask coefficient branches are combined, using a linear combination of the former with the latter as coefficients, followed by a sigmoid nonlinearity to produce the final masks.

The loss function of YOLACT is similar to that of mask R-CNN. The classification, box regression, and mask losses correspond to weights of 1, 1.5, and 6.125, respectively. Both classification and box regression losses are similarly defined as in SSD [104]. Whereas, the mask loss is defined as the per-pixel binary cross-entropy loss of the predicted and ground truth masks.

4.2.2 Field trials based on the YOLACT instance segmentation

The YOLACT was used in a previous study to effectively identify rumen protozoa in microscopic images [105] and large-scale instance segmentation of outdoor environments [106]. To utilise the YOLACT for identifying the harvesting complexity of strawberries, a strawberry dataset comprising 560 images collected by a webcam from the farm under different weather conditions and polygonal annotation, with labels added manually, was used. The YOLACT model used in this study was trained via Google Colab, with a batch size of 8 and iterations of 30,000. Images input into the model were resized to a resolution of 640 × 640. For training, 50 images were selected from the dataset, as shown in Figure 4.13. The YOLACT model achieved excellent results in identifying strawberries with different complexity levels. Overall, this sub-chapter verified that state-of-the-art deep learning models can achieve harvesting complexity detection. Thus, they are integrated with an active system of harvesting robots in the following chapters.


hard: 0.77
hard: 1.00
hard: 0.28
medium: 1.00
hard: 1.00
medium: 0.88
hard: 1.00
hard: 0.99
hard: 1.00
medium: 0.95


easy: 1.00
medium: 1.00
hard: 1.00
medium: 1.00
hard: 1.00
hard: 1.00
hard: 1.00
medium: 1.00


medium: 1.00
hard: 1.00
medium: 1.00
medium: 0.89

Figure 4. 13 - Visualisation of the YOLACT detection results.

Similar to the previous architecture of the GAN-based perception system, the trained model was switched to YOLACT, and the updated perception system framework is shown in the figure below.

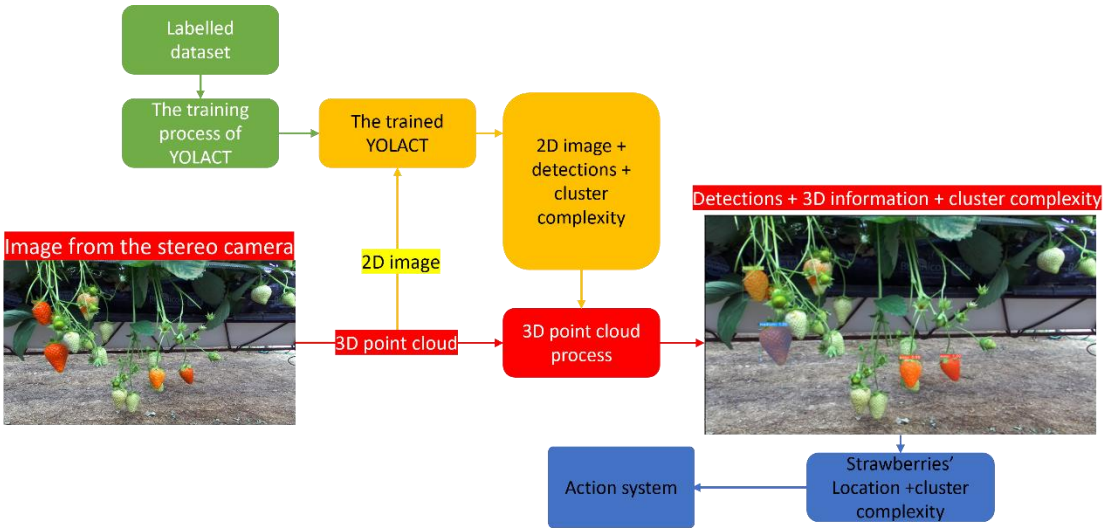


Figure 4. 14 - Overall architecture of the perception system based on the YOLACT.

4.3 Conclusions and Research Directions

This chapter described the development of a perception system for identifying and classifying strawberries. The proposed system uses cGAN (pix2pixHD) trained on synthetically generated data, which incorporated a range of variance in lighting conditions and occlusions as observed in real-world conditions, thus eliminating the need for manual collection and labelling. Such synthetic data can be generated for a range of other crops as well, hence enabling configurability.

However, the detection of strawberries was insufficient for harvesting by robots. Thus, another harvesting complexity-based model was developed to help the robot discriminate between strawberries that are hard and easy to harvest. Unlike the pix2pixHD model, the YOLACT model was selected to determine three complexity levels (i.e., easy, medium, and hard). Although this model can further guide the robot to recognise strawberries that are difficult to harvest, manual labelling is necessary for the training data.

Overall, both pix2pixHD and YOLACT have unique advantages. First, if the robot needs to harvest crops without needing cluster complexity, pix2pixHD can be easily

implemented as a perception system without numerous manual operations. When crops that are fragile and distributed in different cluster complexity need to be harvested, models such as YOLACT with instance segmentation function can be effective for classifying the complexity levels. The contribution of this work is not the use of YOLACT, but the introduction of cluster complexity into object recognition, which allows deep learning models to determine the cluster complexity of objects.

This chapter leaves some interesting ideas for future extensions:

1) Several crop clusters face cluster complexity problems, similar to strawberries. Thus, the proposed approach can be considered not only for soft fruits but also for various cross-industry applications.

2) In the fruit industry, pests and diseases severely affect the yield of fruits. Therefore, identifying rotten strawberries is crucial in commercial farming. Furthermore, the robot may occasionally pick rotten strawberries that are not fit for consumption. Thus, the perception system should have a more advanced classification ability to determine strawberries that are suitable for picking and selling.

3) Although deep learning-based target detection can identify and locate strawberries well, the method cannot accurately determine the ripeness of strawberries. Therefore, hyperspectral imaging-based strawberry ripeness monitoring can be a direction for future work.

Chapter 5

Configurable Action: Task-Specific and adaptive motion control architecture for robotic harvesting

Soft fruits are typically selectively picked manually, as shown in Figure 5.1, as they are small, easily broken, and difficult to pass to traditional machines. However, workers do not prefer this kind of seasonal labour owing to low job skills and tedious work; thus, intelligent agricultural robots are expected to replace this type of work.

Figure 5. 1- Scenery where staff picks strawberries on the farm.

The action of “reaching” is fundamental for agricultural robots to realise the harvesting process. In particular, moving the end-effector accurately toward the fruit and motion planning must be addressed. As discussed in Chapter 2, to shift the cost function to the force field, this study used a neural network implementation of the passive motion paradigm (PMP) based on impedance control and equilibrium point hypothesis for addressing motor control and synergy formation in agricultural robots.

5.1 Passive Motion Paradigm for Goal-directed Reaching

From the perspective of neural control of movement, a PMP network should be considered a “body schema” or an “internal model” that interfaces higher cognitive levels (reasoning and planning) with lower control levels, related to actuators and body dynamics. It is not a controller in the strict sense and thus it is not concerned with dynamics and actuators [65].

5.1.1 Artificial neural network for the internal model of the body

For robot manipulation actions, a novel neural control framework was proposed for goal-directed reaching while considering a range of task constraints. The architecture particularly enables a) swift learning of the internal model of the arm/body and extension to the range of coupled tools; b) runtime incorporation of various task constraints (i.e., end-effector pose, joint limits, tool orientation, motion trajectory, and approach toward the target); c) temporal synchronisation and bimanual coordination for harvesting with two hands; and d) forward simulation of the consequences of action to support goal-directed reasoning. Figure 5.2 shows the block diagram summarising the design of the ANN-based controller from data generation to goal-directed reaching with the Essex agricultural robot. The robot comprises two 6-DoF universal robot 3 (UR3 arm), an ultra-lightweight, compact collaborative industrial robot, and a stereo camera, all housed on a

Husky mobile base. In this work, dual-armed strawberry harvesting is the main task, so it is possible to add task constraints of joint rotation and end-effector pose to achieve collision-free and efficient harvesting motion planning for arms.

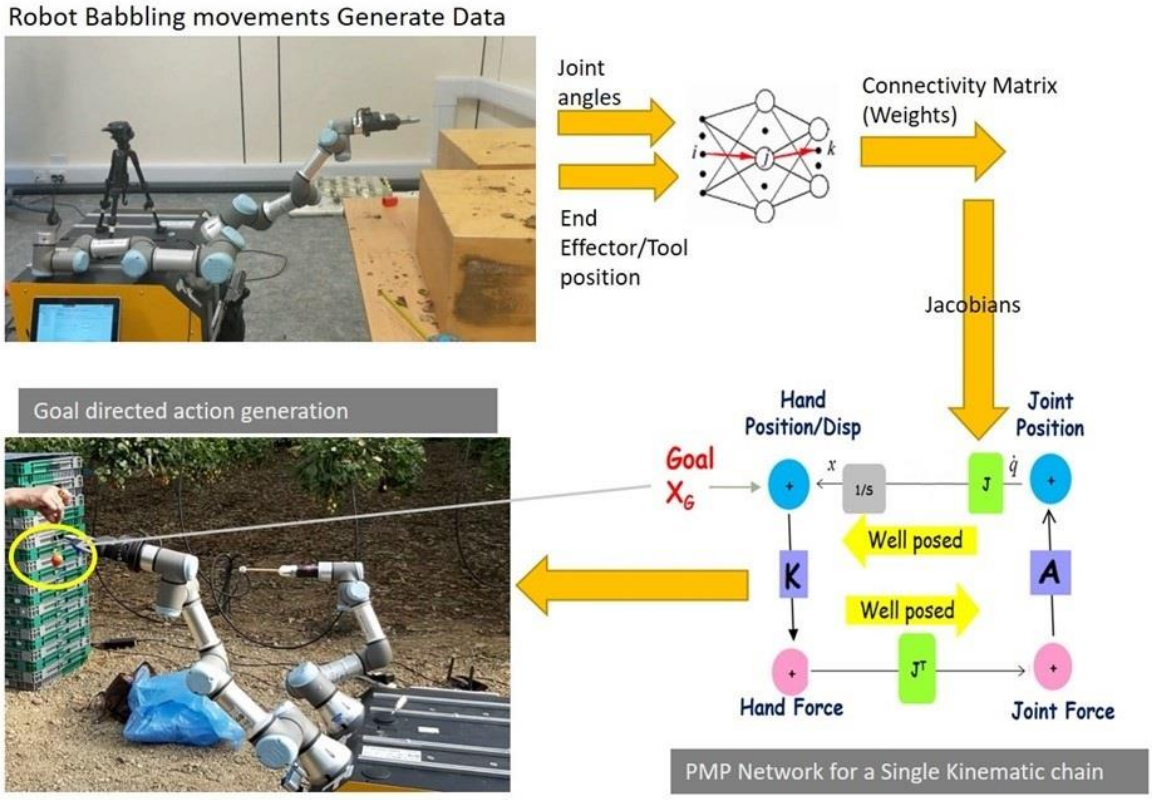


Figure 5. 2 - Artificial neural network based controller begins with the babbling movements of the robot to generate data (top left) which is used to train the backpropagation network (top left). From the connectivity matrix, the Jacobians can be computed (bottom right and Eq. 5.2). The bottom left picture shows the arm reaching the target (X_G).

The PMP computation steps are summarised below. a) *Data generation through robot babbling movements*. The training data for the ANN was obtained through sensorimotor exploration/babbling. In the arms workspace, the UR3's joint rotation readings and set of corresponding end-effector coordinates were saved into two files based on the forward kinematic analysis. The training set comprised 10,000 points in the workspace of the arm and corresponding joint angles.

b) *Design of the neural controller*. Once the training data was obtained, as shown in Figure 5.3, a standard backpropagation network with two hidden layers was used to learn

the mapping $\mathbf{X} = f(Q)$. Here, $\mathbf{Q} = \{q_i\}$ denotes the input vector (of joint angles of the UR3 arm), $\mathbf{X} = \{x_k\}$ denotes the output vector (representing the 3D position/orientation of the end-effector) $\mathbf{Z} = \{z_j\}$, and $\mathbf{Y} = \{y_l\}$ denotes the output of the first and second hidden layer units of the neural network respectively. Eq. 5.1 expresses the mapping, where $\{\omega_{ij}\}$ are connection weights from the input layer to the first hidden layer, $\{o_{jl}\}$ are the connection weights between two hidden layers, $\mathbf{W} = \{w_{lk}\}$ are the connection weights from the second hidden layer to the output layer, $\mathbf{H} = \{h_j\}$ are the net inputs to the neurons of the first hidden layer and $\mathbf{P} = \{p_l\}$ are net inputs to the second hidden layer. Neurons of the two hidden layers fire using the hyperbolic tangent function; the output layer neurons are linear.

$$\mathbf{X} = f(\mathbf{Q}) \Rightarrow \begin{cases} h_j = \sum_i \omega_{ij} q_i \\ z_j = g(h_j) \\ p_l = \sum_j o_{jl} z_j \\ y_l = g(p_l) \\ x_k = \sum_l w_{lk} y_l = \sum_l w_{lk} \cdot g\left(\sum_j o_{jl} z_j\right) \\ \Rightarrow x_k = \sum_l w_{lk} \cdot g\left(\sum_j o_{jl} \cdot g\left(\sum_i \omega_{ij} q_i\right)\right) \end{cases} . \quad (5.1)$$

Concerning the use of external objects as tools, the same procedure can be applied to the data (i.e., end-effector motion and the corresponding consequence on the tool effector) acquired by imitating the teacher's demonstration [68], [107] thus constraining the domain of random exploration.

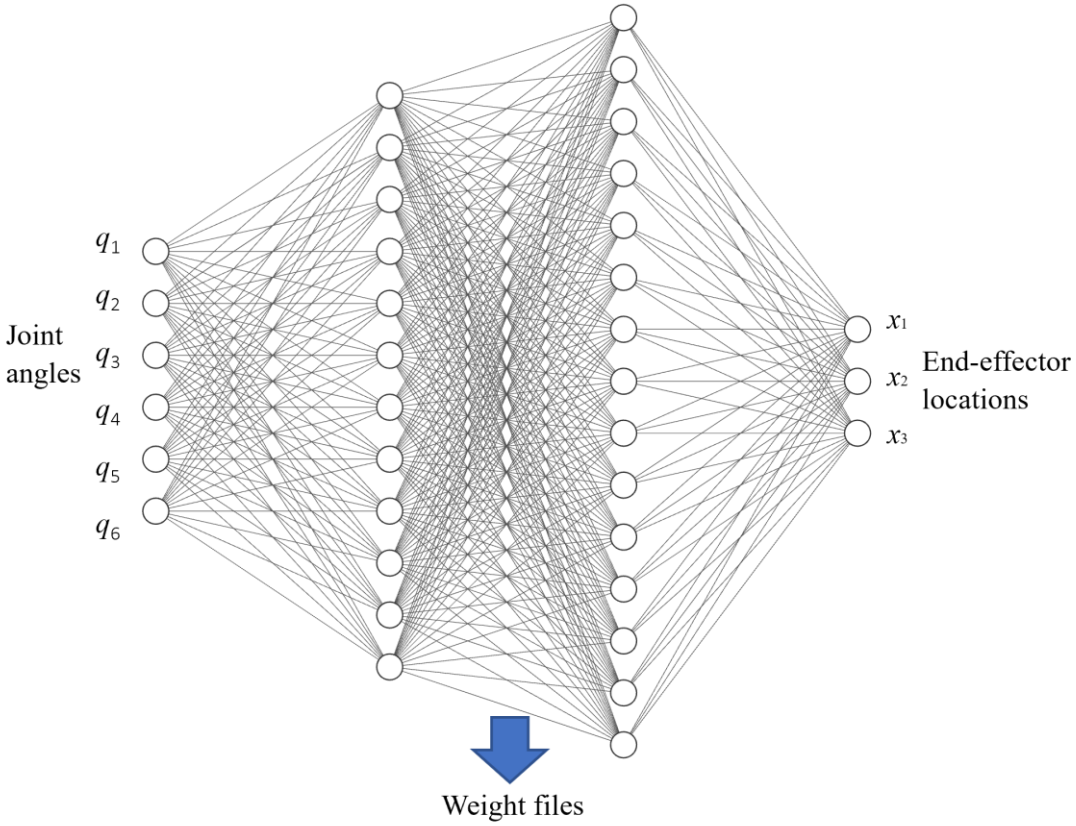


Figure 5. 3 - Backpropagation neural network. The input is the angles of the six joints, whereas the output is the 3D coordinate point of the end-effector. The network is trained to approximate the kinematic transformation and used to evaluate the Jacobian matrix.

The Jacobians encoding the geometric relationship between the respective motor spaces (joint space-end effector space of the UR3 arm) can be extracted from the learning weights of the neural network using the chain rule, as follows (Eq. 5.2).

$$J = \frac{\delta x_k}{\delta q_i} = \sum_l w_{lk} \cdot g'(p_l) \sum_j o_{jl} \cdot g'(h_j)\, \omega_{ij} \,. \tag{5.2}$$

c) PMP network and goal-directed reaching. Once the ANN was trained, the PMP network was generated for goal-directed reaching/control of the arm. The network shown in Figure 5.2 represents the kinematic chain of a single arm. Here, two motor spaces, i.e., hand space with two nodes (representing force (pink) and position of the hand (blue)) and arm joint space with two nodes (representing torque (pink) and rotation of the various joints (blue)), are present. The pair of force–displacement nodes is called a work unit (WU) because the scalar work ($force \times displacement$) is the structural invariant across

different motor spaces. The network can be animated by attaching force fields to one or more body parts/ effectors in a goal-oriented fashion. Animation is analogous to the coordination of a marionette with attached strings (that represent the attractor dynamics of the force field induced by the intended goal i.e. the strawberry). While reaching is the simplest case with a fixed point attractor (at the target), the body schema can be animated with moving point attractors to produce diverse spatiotemporal trajectories, as shown for drawing [68], tool use [107], [108]. The computational model can be summarised as follows.

Let $\mathbf{q}$ denote the set of all the DoFs that characterise the UR3 arm. Subsequently, the kinematic transformation $\mathbf{x} = f(\mathbf{q})$ can be expressed as: $\dot{\mathbf{x}} = J \cdot \dot{\mathbf{q}}$ where J is the Jacobian matrix of the transformation extracted from the trained ANN. Next, the PMP animation in the simplest case for a serial kinematic chain involves the following steps.

(1) *Generate a target-centred, virtual force field in the extrinsic space*:

$$\mathbf{F} = K_{ext}(\mathbf{x}_G - \mathbf{x}), \tag{5.3}$$

where $\mathbf{x}_G$ denotes the strawberry to reach and K_{ext} is the virtual stiffness of the attractive field in the extrinsic space. K_{ext} determines the shape and intensity of the force field. In the simplest case, *K* is proportional to an identity matrix and this corresponds to an isotropic field, converging to the target along straight flowlines.

(2) *Map the force field from the extrinsic space into the virtual torque field in the intrinsic space*:

$$\mathbf{T} = J^T \mathbf{F}. \tag{5.4}$$

(3) *Relax the arm configuration to the applied field*:

$$\dot{\mathbf{q}} = A_{\text{int}} \cdot \mathbf{T}, \tag{5.5}$$

where A_{int} denotes the virtual admittance matrix in the intrinsic space. The modulation of this matrix affects the relative contributions of the different joints to the overall reaching movement.

(4) *Map the arm movement into the extrinsic workspace*:

$$\dot{\mathbf{x}} = J \cdot \dot{\mathbf{q}} . \tag{5.6}$$

(5) *Integrate over time until equilibrium*:

$$\mathbf{x}(t) = \int_{t_0}^{t} J\dot{\mathbf{q}} d\tau . \tag{5.7}$$

The fifth step is integration, which provides a trajectory with the equilibrium configuration $\mathbf{x}(t)$ defining the final position of the robot in the extrinsic space. All the computations in the above loop are "well-posed" and the relaxation mechanism does not require any cost function to be specified to solve the indeterminacy related to the excess DOFs (the redundancy problem). Time can be explicitly controlled by inserting a time-varying gain $\Gamma(t)$ in the nonlinear dynamics of the relaxation process (Eqs. 5.3–5.6). To achieve this, the technique originally proposed in [109] for content addressable memories can be extended in the context of goal-directed reaching for robots and used [110].

This can be implemented by substituting the relaxation Eq. (5.5) with Eq. (5.8), as follows:

$$\dot{\mathbf{q}} = \Gamma(t) \cdot A_{\text{int}} \cdot \mathbf{T}, \tag{5.8}$$

where a possible form of time-varying gain is the following that uses a minimum-jerk generator with duration *t*.

$$\Gamma(t) = \frac{\dot{\xi}}{1-\xi}, \tag{5.9}$$

where

$$\xi(t) = 6(t/\tau)^5 - 15(t/\tau)^4 + 10(t/\tau)^3 . \tag{5.10}$$

In general, a time base generator (TBG) can be used as a computational tool for synchronising multiple relaxations in composite PMP networks, essentially coordinating the relaxation of movements of two arms, or even the movements of two robots.

For a simple reaching task with an arm, at the end of the animation process, four sets of trajectories are obtained as a function of time: 1) sequence of joint angles given by the positioning node in the joint space (arm); 2) resulting consequence i.e. the sequence of end-effector position given by the positioning node in end-effector space; 3) sequence of torques at the different joints (arm and waist), given by the force node in the joint space; 4) resulting consequence i.e. the sequence of forces applied by the end-effector given by the force node in the end-effector space. The time-varying gain is considered a temporal pressure that becomes stronger as the deadline approaches and diverges afterward. Further details of the mathematical model for terminal attractor dynamics applied to goal-directed reaching in robots can be found in [110].

Simultaneously, a range of internal and external constraints can be integrated at runtime based on the requirements of the task that needs to be performed as force fields defined either in the extrinsic space or in the intrinsic space.

5.1.2 Spatial planning for bimanual manipulation

This study focused on developing a perception–action decision system for the dual-arm mobile robot harvesting strawberries in the greenhouse. Therefore, for this application, the basic PMP sub-network (Figure 5.2) was repeated for the right and left arms. The bimanual coordination task of reaching two objects simultaneously is shown in Figure 5.4. The network shown in Figure 5.4 (a) represents the kinematic chain of the dual arm. In normal conditions, all the participating joints were considered equally compliant. Here, the admittance A_{int} is an identity matrix (for UR3, it is a 6×6 identity matrix). However, by locally modulating individual values, the degree of participation of

each joint in the coordinated movement can be varied while maintaining the solution at the end-effector space. Here, a specific constraint for the shoulder joint of each arm was applied to avoid arms collision. This is because when the first joint (shoulder) of the robotic arm rotates, the upper and lower arms exhibit large movements. Decreasing the degree of participation of the shoulder is an effective way to avoid collisions.

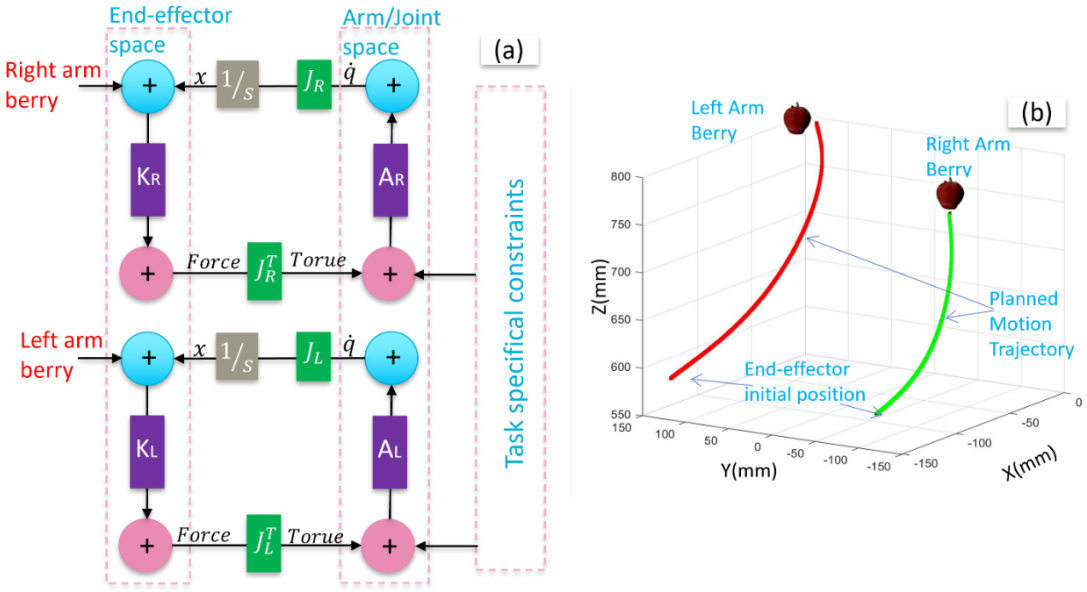


Figure 5. 4 - (a) Dual-arm passive motion paradigm network model for fruit harvesting; (b) Example of planned motion trajectory based on given strawberries' positions.

Similar to a reaching task with one arm, the planned trajectories for a bimanual coordination task are illustrated in Figure 5.4 (b) at the end of the animation process.

5.2 Analysis of the Action System

5.2.1 Accuracy analysis

To verify the action system, an example of results when PMP is provided with a target to reach is presented below. Figure 5.5 shows the harvest process of strawberries in the laboratory. Figure 5.6 (a) shows the transition from the initial position to the end-effector's final target position. Similarly, Figure 5.6 (b) shows the sequence of arm joint angles in all DoF from its initial position to its final position for the end-effector to reach

the target. The results were as expected, within a few millimetres of the target set. A key observation was the smoothness of the curves in the figures reflecting the framework's natural no-jerk behaviour. Finally, in Figure 5.6 (c), the graph shows the system's time pressure to finish arm movement in the set number of iterations (i.e., 1000). Figure 5.7 illustrates the simulation results from the MATLAB. In this simulation process, 200 target points (black) were randomly generated in the arms working space. Then the PMP calculated the arm's joint angles with the end-effector's corresponding position (green point). The target points, therefore, have an average error of 2.8853 mm compared to the positions of the end-effector; here, some black dots are not visible as they are covered by green dots.

Figure 5. 5 - Test of the action system in a lab setting: (a) arm reaches the target position, (b) gripper cuts the stem of the target strawberry, and (c) gripper remains closed until it goes to the specified position.

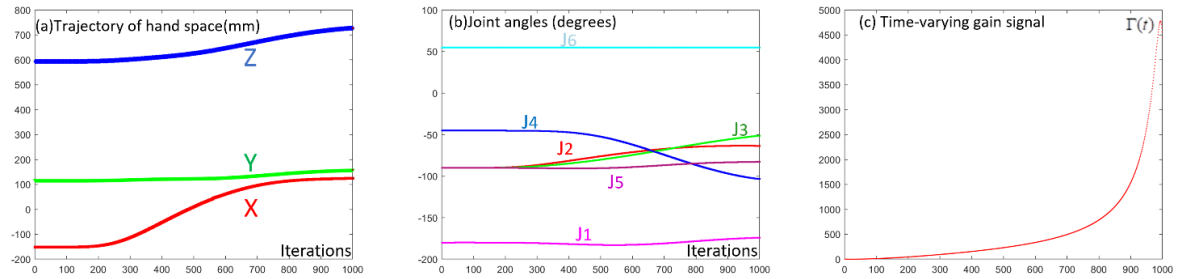


Figure 5. 6 - a) Sequence of end effector position from an initial position (-151, 116, 593) to the target (124, 158, 727) as a function of time; b) Sequence of joint angles in all the DoF of the arm from an initial state to the final state (when the end effector reaches the goal); (c) Time-varying gain signal.

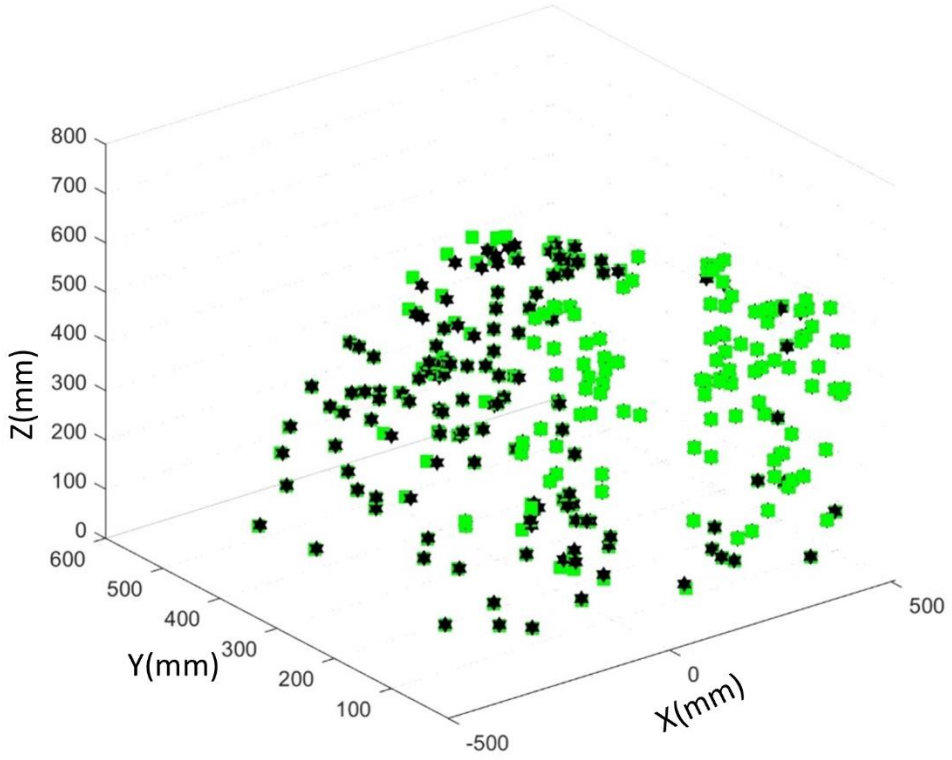


Figure 5. 7 - Target reaching accuracy for 200 points in the workspace. The black points are the target locations and the green points are the results given by the PMP. Some of the targets are completely covered by the green points; hence, they cannot be displayed.

5.2.2 Harvesting speed analysis

Joint acceleration and speed of the leading axis are the key parameters for the execution time. To illustrate one arm harvesting time, the acceleration and speed were first set to 4 rad/sˆ2 and 4 rad/s, respectively. Note that when the gripper opened or closed, a delay of 0.8 s was added in the execution of the program. Thus, the average time of the entire execution process of single strawberry harvesting (i.e., strawberry detection, mobile base movement, arm movement, and placing strawberry) was approximately 11 s, as shown in Figure 5.8. In this figure, despite the recognition results of the perception system, a timer is present in the upper right corner to count the running time of the system, whereas the lower left

"*You cannot have the best of both worlds.*"

Speed and accuracy are often issues that cannot be combined perfectly.

corner shows the real-time 3D coordinates of each strawberry. When the acceleration and speed were increased to 20 rad/sˆ2 and 20 rad/s, respectively, the average time of the execution process of three strawberries harvesting was approximately 19 s, as shown in Figure 5.9.

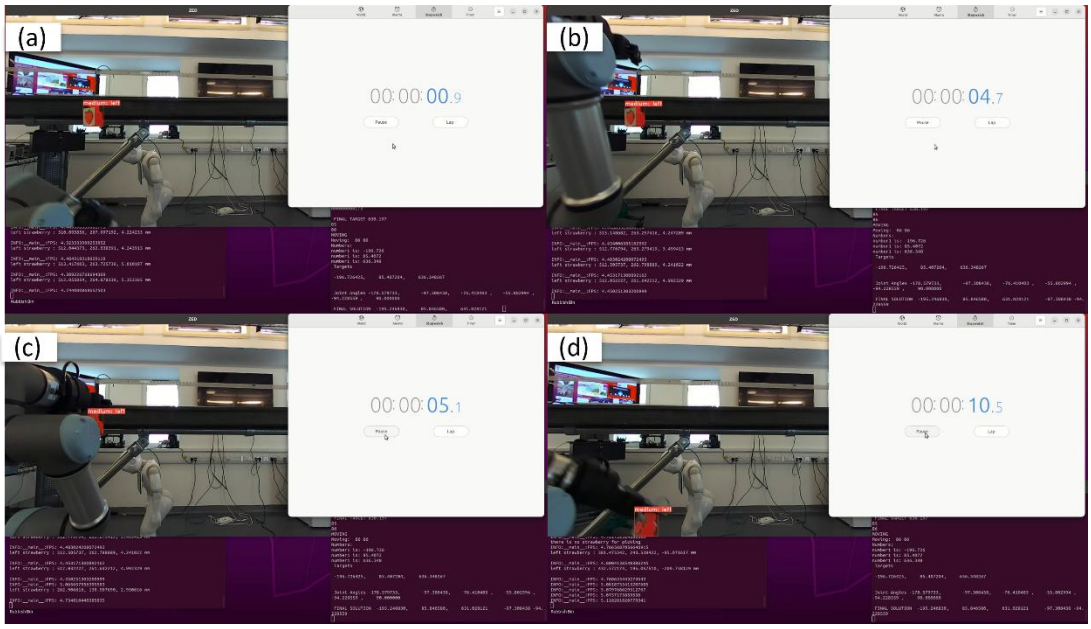


Figure 5. 8 - Single strawberry harvesting with low speed. (a) Strawberry detection and mobile base movement; (b) Arm movement; (c) Gripper working; (d) Placing strawberries.

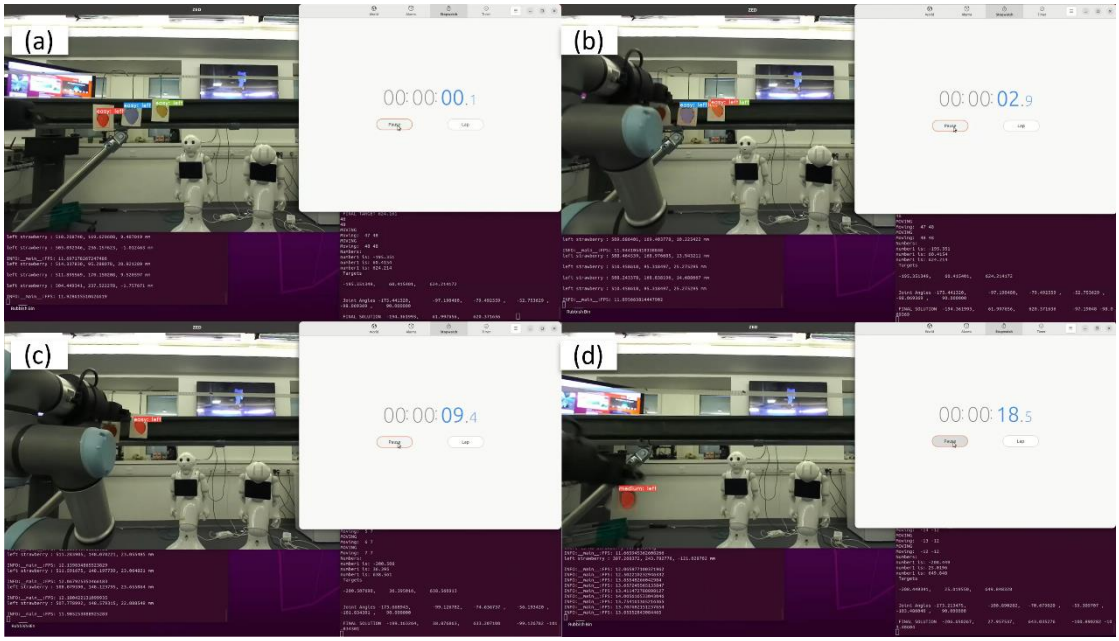


Figure 5. 9 - Three strawberry harvesting with high speed. (a) Strawberry detection and mobile base movement; (b) First strawberry harvesting; (c) Second strawberry harvesting; (d) Placing the last strawberry.

If the speed is greater than 20 rad/s, the robotic arm may vibrate, thus affecting the regular operation of the robot. Evidently from the above high-speed testing, the speed of the robot was close to that of a human picker (3.5–5.0 s for searching and picking one strawberry). However, in the field, the average time may be influenced by the uneven ground and distribution of strawberries. Figure 5.10 illustrates the harvesting speed on the farm with medium speed (i.e., the acceleration and speed are set to 15 rad/sˆ2 and 15 rad/s, respectively). As shown in Figure 5.10, a single robot arm takes approximately 14 s (including the movement time of the mobile base) to pick two strawberries in a row on the farm.

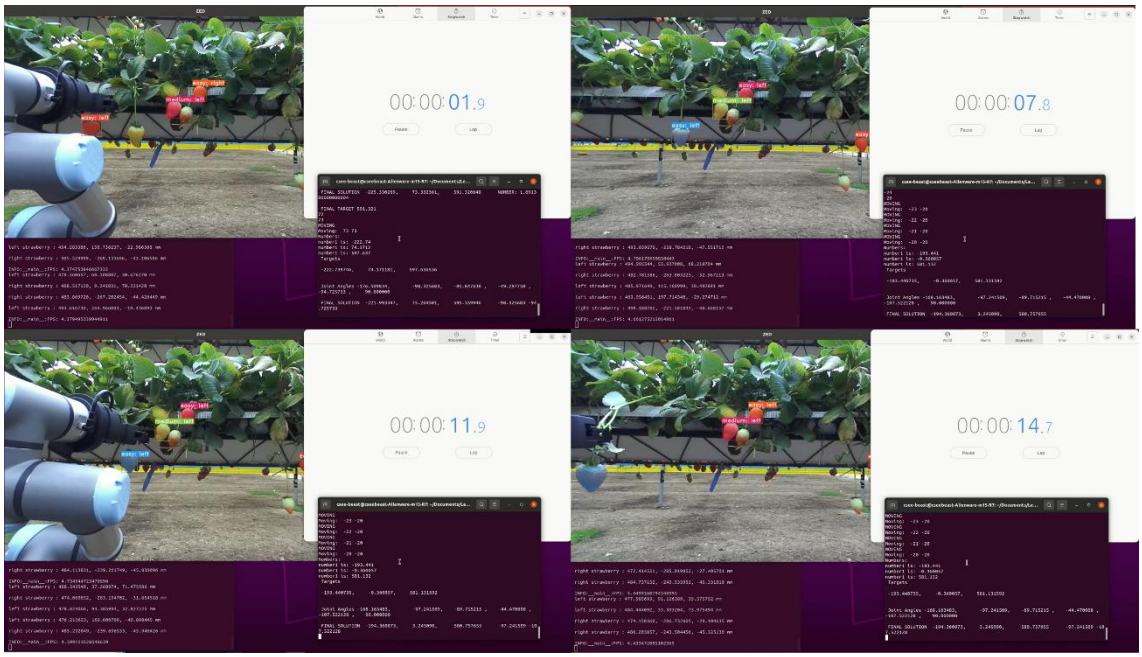


Figure 5. 10 - Strawberry harvesting with medium speed on the farm. A single arm spends approximately 14 s (including the time of mobile base movement) to pick two strawberries.

5.3 Results from Field Trials

To demonstrate the working of the proposed robotic action system in real-world environments, field experiments were conducted in 2021–2022 in the vertical greenhouse in Tiptree Essex, UK. As shown in Figure 5.11, strawberries were grown in a vertical system such that the strawberry table-tops could be raised up or lowered. This field trial

round focused on whether the robotic action system can pick low-harvesting complexity strawberries. More details regarding the robotic perception–action system for bimanual manipulation are discussed in the next chapter.

Figure 5. 11 - Layout of the vertical greenhouse. The strawberry table tops can be raised or lowered, thus providing a passable aisle for robots and staff.

During harvesting, the perception system first obtains the 3D information of the target strawberry. Subsequently, the mobile base determines whether it must move horizontally according to the distance to the strawberry. Finally, the action system calls the PMP for harvesting. In detail, the gripper cuts the stem of the strawberry, as shown in Figure 5.12, as follows. Once the robot acquires the central location of the strawberry, the

perception system estimates the cutting point based on the bounding box. Next, it opens its gripper and advances to the point. Thus, damage to the strawberry is avoided by cutting only the stem; meanwhile, once the gripper is open, the area to cut the stem is sufficient, even if the strawberry's stem is curved. Figure 5.12 (b) illustrates the geometric relationship between the bounding box (the blue box in the figure) and the cutting point. Once a strawberry is detected, two corner points are defined, and the cutting point is readily estimated based on the two points. The perception system calculates the top centre of the detected strawberry first in 2D pixels and then converts it to a 3D point based on the point cloud. When the top centre of the strawberry's location is finalised, the expected cutting point is estimated as 2 cm above the top centre of the strawberry. Figure 5.12 (c) shows an example of the gripper cutting the stem of a strawberry.

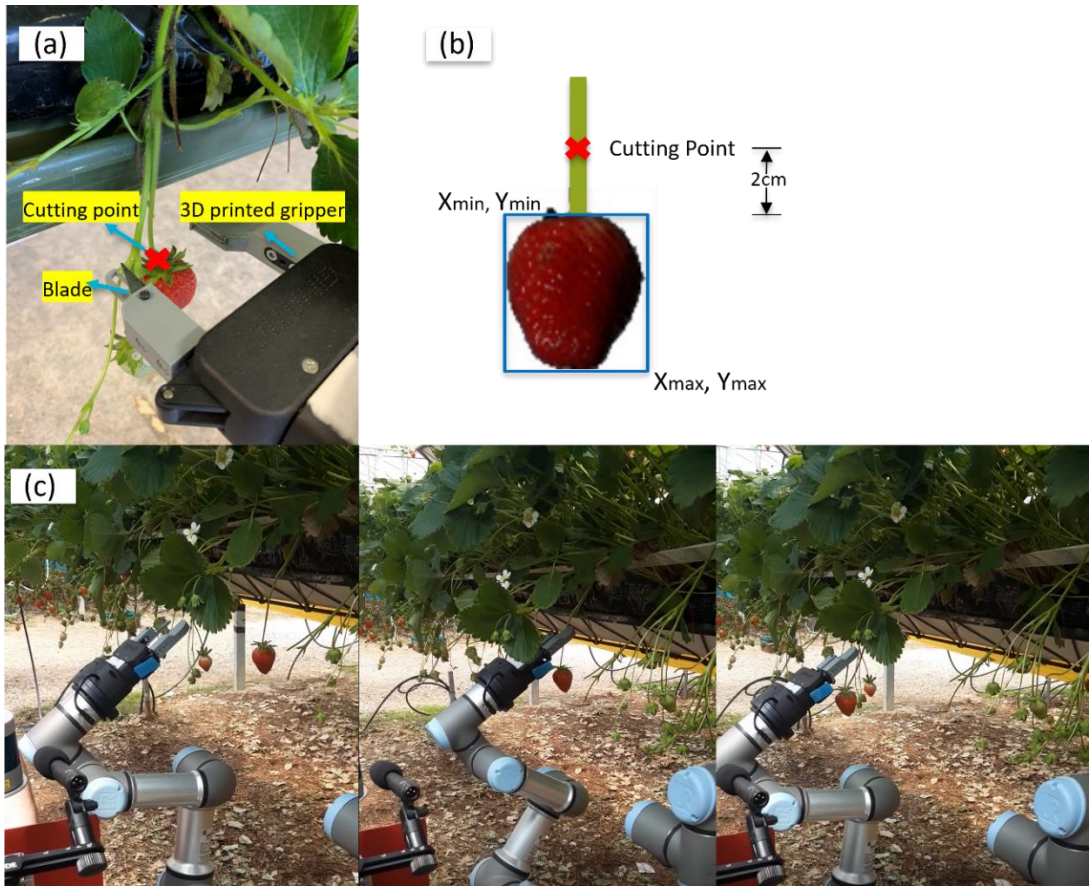


Figure 5. 12 - (a) Gripper/cutter of the robot; (b) Geometric relationship between the bounding box and cutting point; (c) Reaching and cutting a target in the field.

In the field experiment, the robot attempted 35 pickings, of which 11 were failed. Thus, it exhibited a successful picking rate of 68.57 %; success and failure cases are shown in Figure 5.13. One of the main reasons for failure was the short size of the blade in the gripper, which could not cut the stem accurately, thus resulting in pulling or fruit falling. Additionally, the position error (i.e., calibration of the camera, coordinates transformation between camera and arm base) was another factor that led to inaccurate picking. In order to improve position errors as much as possible, the camera is first fixed to the robot to ensure a stable position relationship between the camera and the robot arm. A calibration algorithm is then used to transfer the coordinates from the camera to the arm base, which will be introduced in the next chapter.

Additionally, the harvesting performance depends on whether the target strawberry is surrounded by obstacles (immature berries) or whether its stems are entangled. To tackle this, the fruit cluster complexity analysis was introduced in Chapter 4.2. Generally, because each cluster's complexity is uncertain and random, realising the autonomous harvesting of the greenhouse is still a challenge for the robot. The next chapter addresses this challenge based on field experiments.

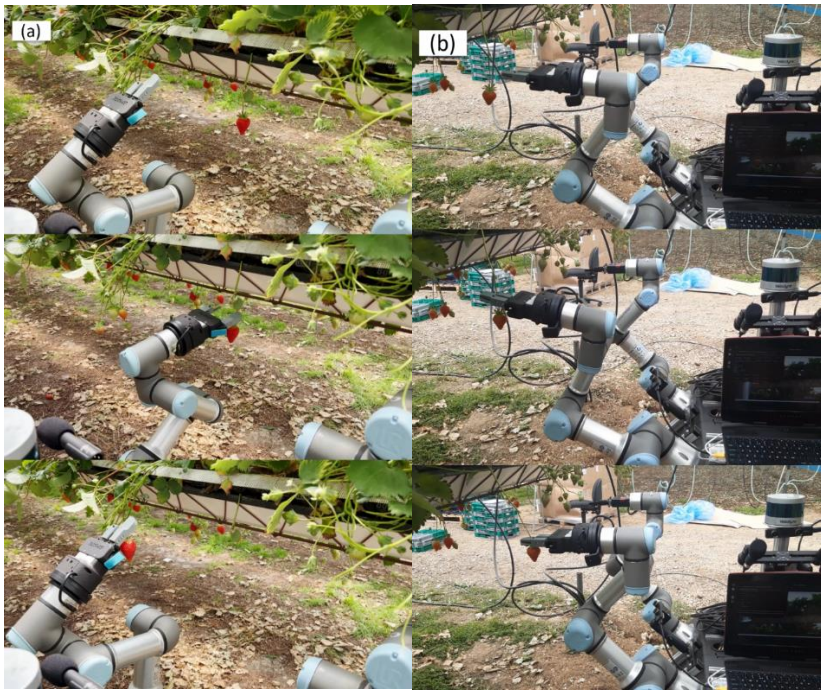


Figure 5. 13 – Field trials. (a) Success case; (b) Failure case: cannot cut the stem(“pulling”); (c) Failure case: position error.

5.4 Summary

This chapter presented a biologically inspired action system for robotic soft fruit harvesting, and a PMP for goal-directed reaching with a mean error of less than 3 mm in the laboratory. This framework was field-tested in a state-of-the-art vertical growing system at Wilkin and Sons, Tiptree, Essex. The action system is a forward/reverse model that can be used to simulate the consequences of predictive planning and to extend a series of tools coupled with the arm. Compared with the conventional optimisation control

method, this method can effectively solve the DoF problem and realise the high-precision movement of robotic arms. The results illustrated the overall performance of the action system and the smooth harvesting process.

However, given the substantial variance in the structure of the canopy, the integration of cluster complexity analysis and bimanual coordination is crucial for the robot to harvest strawberries automatically. In addition to the identification and localisation of the strawberry, this features assigns a complexity level to every identified strawberry. Thus, the complexity level enables planning the strategy for picking, such as reaching with single-arm, body movement, and two-handed coordination (decluttering the obstacle with one hand and picking with the other one). Thus far, the cluster complexity analysis and PMP-based action system have been developed separately; the next chapter integrates them into a dual-arm robot architecture for strawberry harvesting in the field.

Chapter 6

Integration: Perception-Action-Decision Making Loop Targeting Harvesting of Fruits

Recently, robots have been widely used in industries. However, farms are typically unstructured environments; thus, robots are not easily commercialised in agriculture. This chapter focuses on integrating all the previous sub-systems into the Essex agricultural robot and working toward field applications. The robot constructs an action plan based on the current scene, which helps the robot decide which strawberry can be picked by which arm, how much the mobile base must move, and whether the strawberry can be picked successfully without damage according to the complexity level. Herein, the system was extended to a dual-arm mobile robot and its harvesting performance in the vertical greenhouse was verified.

6.1 How Acting Can Make the Robot See Better

As humans, our visual environments comprise multiple objects in everyday life. However, we are constrained in the number of actions we can simultaneously perform owing to limited effector systems. To survive in such environments, we must be able to select stimuli for actions that are of prime relevance to our behavioural goal [111]. Similarly, for a robot with a more limited function of vision and motion, reasonably cooperating with the perception–action system is essential to perform basic tasks. In particular, in this study, the interaction of perception and action in harvesting robots was observed.

In Figure 6.1 a, the robot obtains the goal information from its perception system; its arm reaches the goal at awkward angles, and thus, may damage the target. However, if the robot can combine the mobile base and arm movement, the situation changes. As shown in Figure 6.1 b, the robot moves a short distance to ensure the perception system can detect the goal in front of it. Subsequently, the arm can reach the target smoothly and the gripper can cut the stem in the horizontal direction.

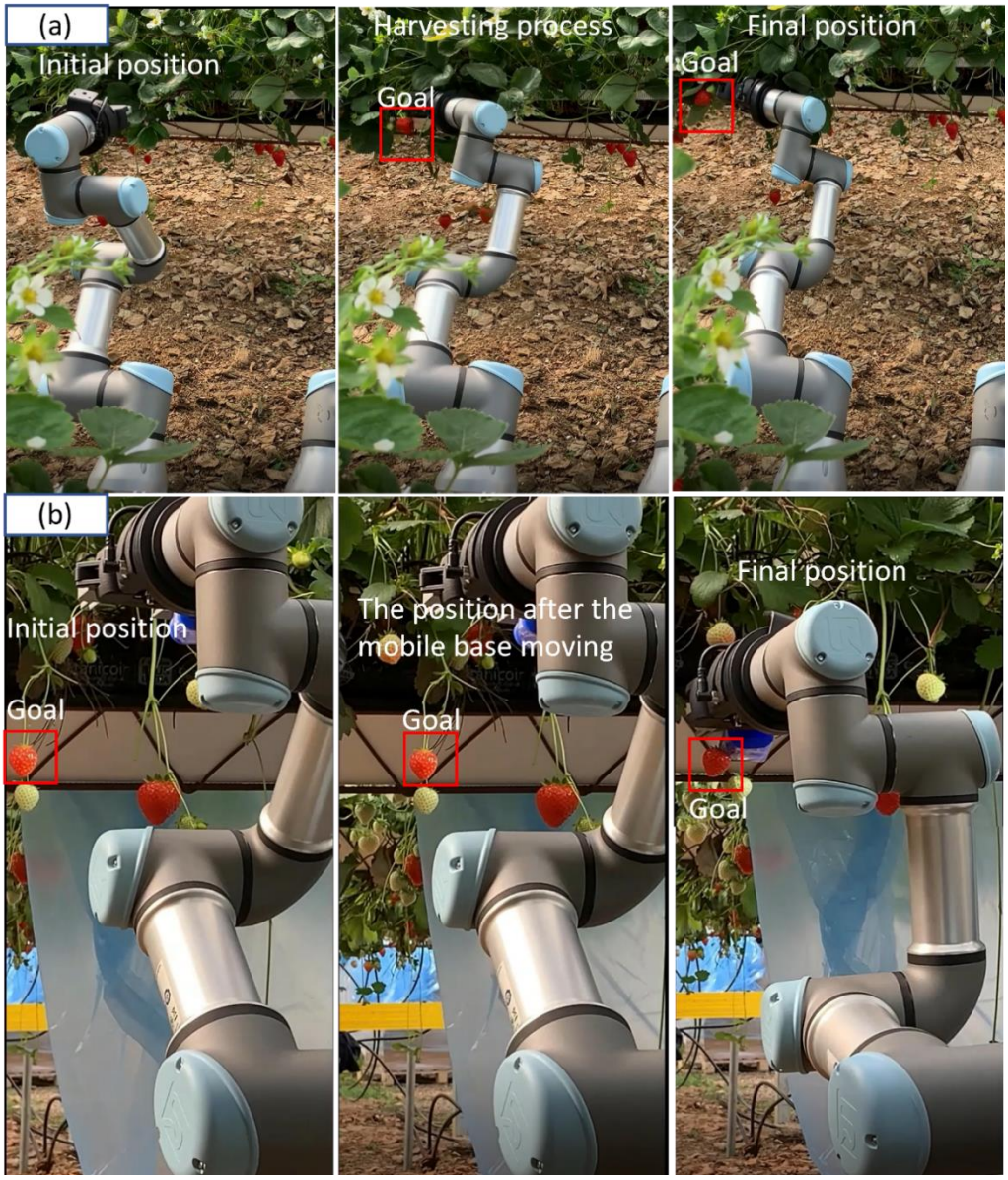


Figure 6. 1 – (a) Reaching a target without the mobile base movement (joint angles look slightly complex). (b) Reaching a target with the mobile base movement (the joint angles do not require significant rotation).

Essentially, a better perspective can be obtained by adjusting the position and pose. However, with a clear behavioural goal, the goal can be reached with a series of simple movements. To illustrate the robot perception–action system, the overview of the system architecture, comprising the user interface, perception system, navigation (mobile base), and robotic arm control, is shown in Figure 6.2.

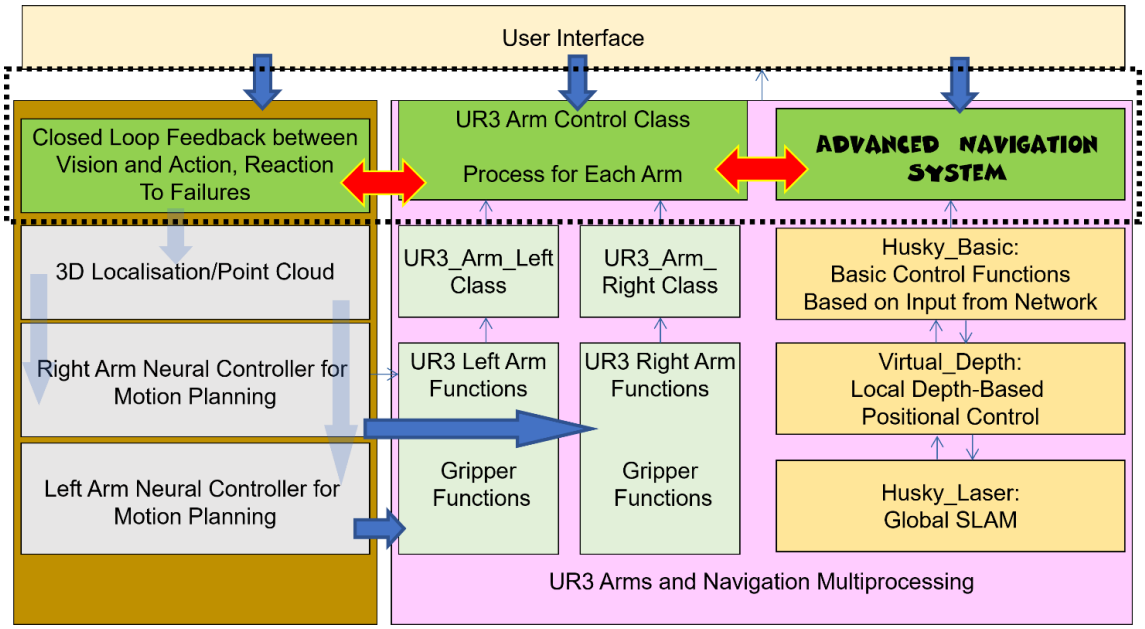


Figure 6. 2 - Overview of the Robotic system architecture. The perception system, action system, and navigation system communicate with each other to transfer data. Each system can be opened individually in the user interface or the entire system can be run with one click.

The proposed system was applied to the Essex agricultural robot, as shown in Figure 6.3a, for laboratory and field experiments. To simplify the use of the system, the user interface provides several selections to initialise the robot, launch the perception system, navigation, and harvest strawberries via both/single arms.

Some positional errors and failure cases were observed in the last season's field experiments (action system testing in the previous chapter). Therefore, through several trials on the farm, the camera position was fixed to an optimal position with a good perspective (Figure 6.3 b). In addition, the structure of the gripper was slightly modified to ensure the stem was cut properly. As shown in Figure 6.3 c, to avoid the “pulling”

cases, a rod was placed to block the fruit stem in the cutting area. More details regarding how the integrated system works are introduced in the rest of the chapter.

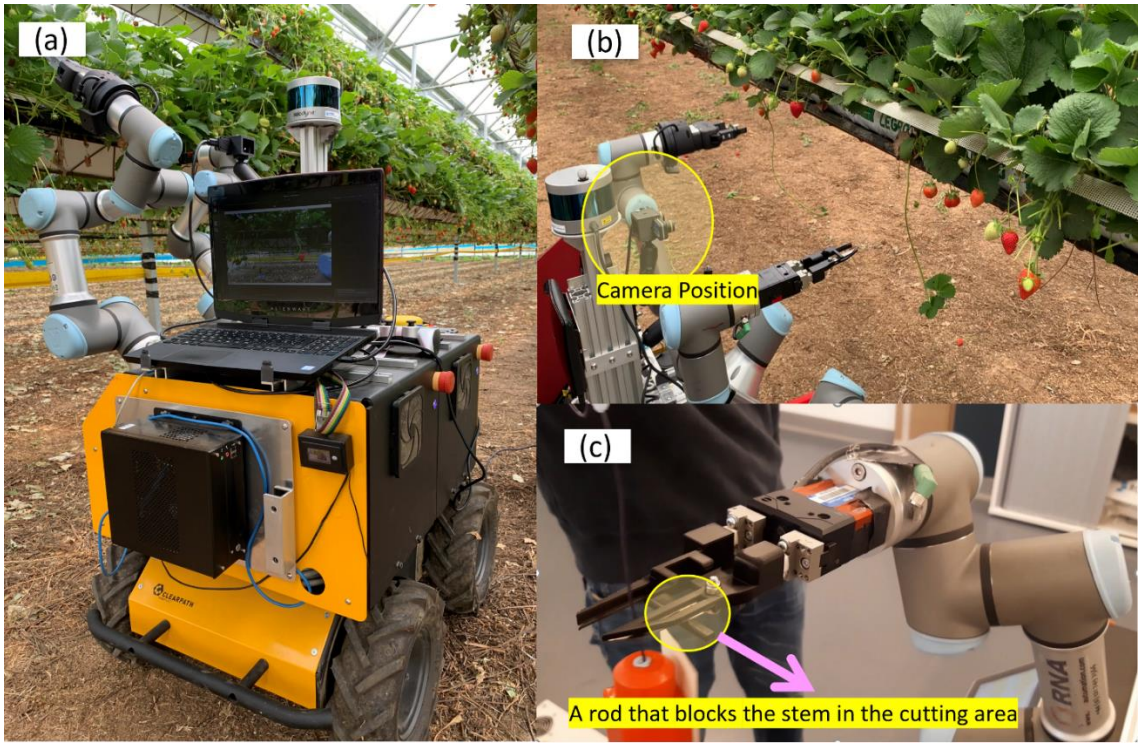


Figure 6. 3 - (a) Essex robot working in the field; (b) Optimal camera position; (c) Updated structure of the gripper.

6.2 Strawberry Allocator: A Forward Action Planner for Bimanual Manipulation

6.2.1 Coordinate transformation

To provide a harvesting/action plan for the dual-arm mobile robot, the perception system must provide the 3D coordinate (x, y, z) and complexity level that can be used to execute this plan. Therefore, the YOLACT-based (see chapter 4.2) detector was selected as the main model to detect and classify the strawberries. The localisation process was realised using a stereo camera. Figure 6.4 shows an example of collecting the 2D and 3D data simultaneously from the farm using the stereo camera.

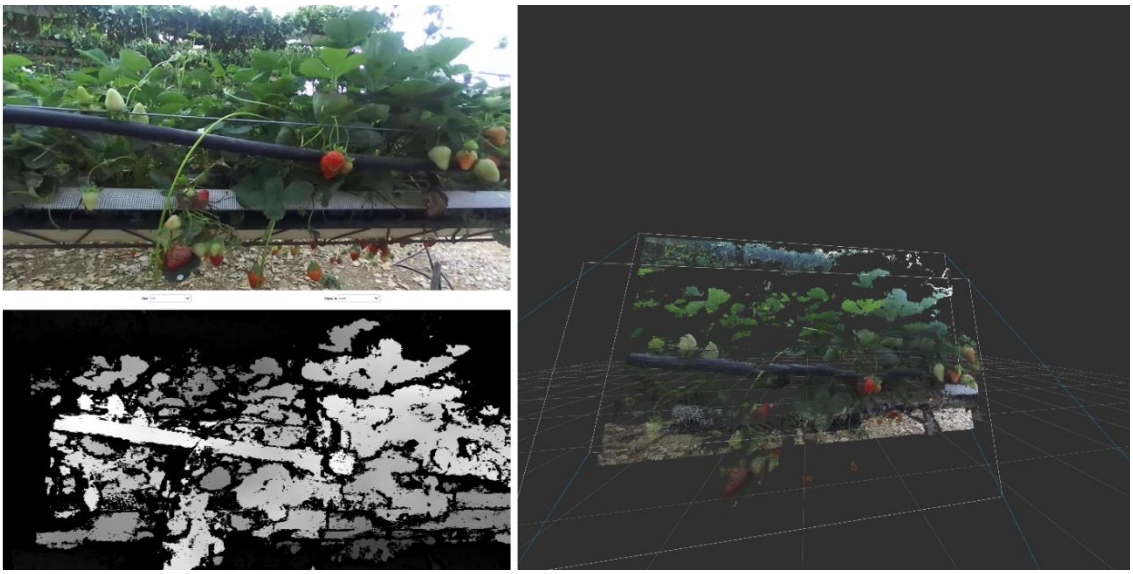


Figure 6. 4 - Obtaining 2D and 3D information for the dataset using a stereo camera.

Furthermore, to decrease the positional error, an algorithm based on the least-squares fitting [74] was selected to determine the optimal solution of rotation matrix R and translation vector for coordinate transformation (Figure 6.5).

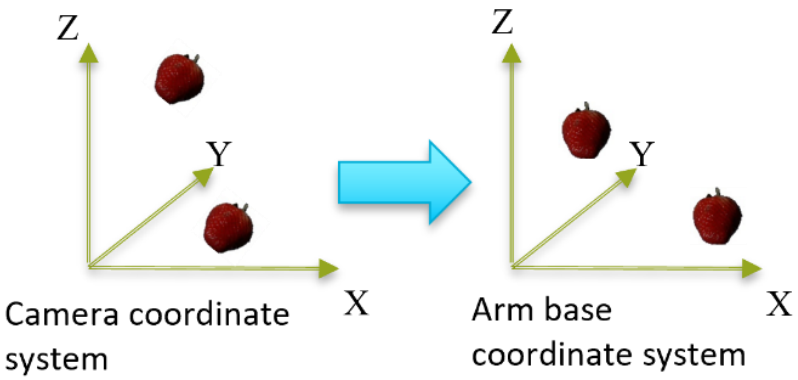


Figure 6. 5 - Illustration of coordinate transformation.

Mathematically, when the stereo camera acquires the 3D coordinate of a strawberry P_{cam} it should be transferred to the arm base coordinate system for inverse kinematic using the following equation.

$$P_{arm} = RP_{cam} + \vec{t}\,. \tag{6.1}$$

To determine the optimal rotation and translation, the point data $\mathbf{P}_{cam} = \left[P_{cam}^{1}, P_{cam}^{2}, ..., P_{cam}^{n}\right]$ and corresponding data $\mathbf{P}_{arm} = \left[P_{arm}^{1}, P_{arm}^{2}, ..., P_{arm}^{n}\right]$ was

collected; subsequently, the solution was computed by minimising the least squares error of the datasets, as follows.

$$err = \sum_{i=1}^{n} \left\| RP_{cam}^{i} + \vec{t} - P_{arm}^{i} \right\|^{2} . \tag{6.2}$$

First, the centroids of both datasets were calculated as follows.

$$\begin{aligned} centroid_{cam} &= \frac{1}{n}\sum_{i=1}^{n} P_{cam}^{i} \\ centroid_{arm} &= \frac{1}{n}\sum_{i=1}^{n} P_{arm}^{i} \end{aligned} . \tag{6.3}$$

Next, singular value decomposition (SVD) [112] was used to determine the optimal rotation, as follows.

$$\begin{aligned} H &= (\mathbf{P}_{cam} - centroid_{cam})(\mathbf{P}_{arm} - centroid_{arm})^{T} \\ [U, S, V] &= \mathrm{SVD}(H) \\ R &= VU^{T} \end{aligned} . \tag{6.4}$$

Finally, translation was obtained, as follows.

$$\vec{t} = centroid_{arm} - R \times centroid_{cam} . \tag{6.5}$$

6.2.2 Strawberry allocator

Once the 3D coordinates and complexity levels of the detected strawberries are acquired, the robot must decide which arm is suitable for harvesting, and how much the mobile base must move. Figure 6.6 illustrates this process by showing four strawberries ready for harvesting.

When multiple harvest-ready strawberries are in view, are they picked at random or in a specific order? Which strawberries can be picked with one hand, and which ones require both hands?

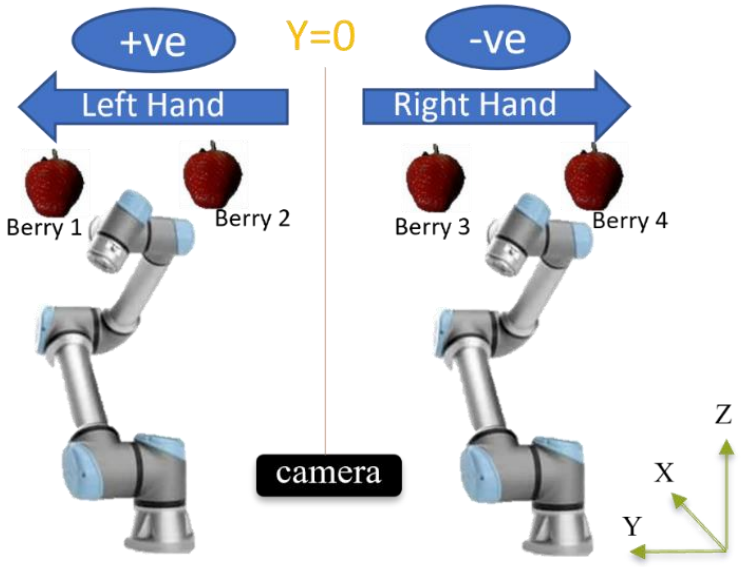


Figure 6. 6 - Working space for the left and right arms. The space is divided based on the Y-axis of the coordinate system, and the camera is located at the origin of the Y-axis, with the left hand on the positive half-axis and the right hand on the negative half-axis.

Once the perception system detected the four berries, the output result is as follows:

$$\begin{cases} berry1\text{: left, } (x_1, y_1, z_1), \text{ easy} \\ berry2\text{: left, } (x_2, y_2, z_3), \text{ easy} \\ berry3\text{: right, } (x_3, y_3, z_3), \text{ easy} \\ berry4\text{: right, } (x_4, y_4, z_4), \text{ easy} \end{cases}$$

The left/right information is obtained based on the coordinate value (x_i, y_i, z_i). When the strawberry is in the middle of the working space, the coordinate y_i value fed back by the stereo camera is zero. When the strawberry is distributed on the left/right side, the value is increased/decreased. To arrange the harvesting sequence, the working space is divided into two areas, for the left and right arms separately. When the robot moves in the direction from left to right in the above figure, each arm first harvests the left-most strawberry in its working area. Conversely, the robot can be set to harvest strawberries from right to left as the robot moves from right to left. By using this picking strategy, collisions between the arms can be avoided and efficiency maintained. That is, as shown in Figure 6.6, the left arm first harvests *berry 1*, and the right arm harvests *berry 3*. Next,

the mobile base moves a short distance and harvests the remaining two berries (see Figure 6.7).

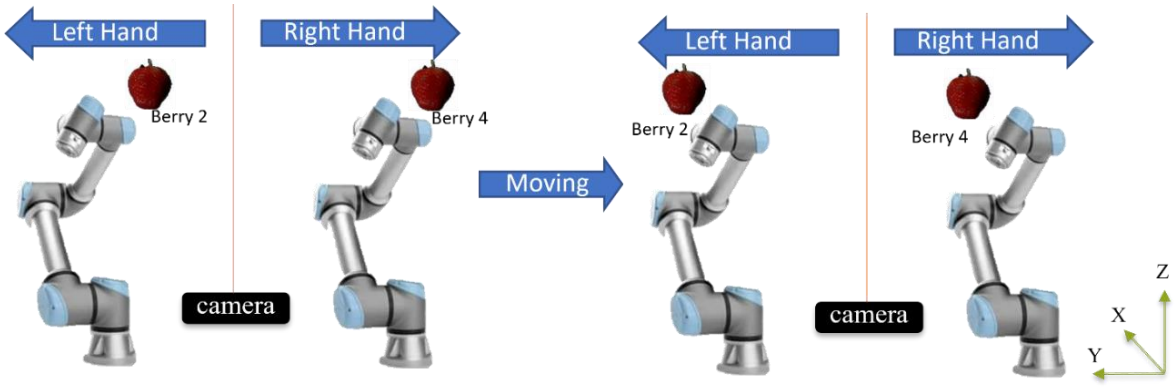


Figure 6. 7 - Harvesting the remaining two strawberries in combination with the moving base movement.

The robot needs to move only in the horizontal direction because the strawberries are all distributed on one side. To calculate the distance the robot must move, its movement was set such that at least one strawberry was in front of one of the robotic arms for harvesting. For example, when the robot moved in the direction shown in Figure 6.7, the distance of the movement ensured that *berry 2* was in front of the left robotic arm for harvesting. This is expressed as follows.

$$\begin{cases} berry2:\ \text{left},\ (x_2, y_2, z_3),\ \text{easy} \\ berry4:\ \text{right},\ (x_4, y_4, z_4),\ \text{easy} \end{cases} \rightarrow \vec{s} = a - y_2 \rightarrow \begin{cases} berry2:\ \text{left},\ (x_2, a, z_3),\ \text{easy} \\ berry4:\ \text{right},\ (x_4, y_4', z_4),\ \text{easy} \end{cases},$$

where $\vec{s}$ denotes the displacement that the robot must be moved. Before the robot moves, the coordinate Y value of *berry 2* is y_2. When the strawberry is located in front of the left arm, its coordinate value becomes *a*. Hence, the moving distance must be calculated according to the constant value *a*. This method is advantageous because if a strawberry is always in front of one arm to harvest, each joint does not need considerable rotation; thus, collisions between components can be easily avoided.

In addition to the above information for the action sequence, the complex level is another factor. In the above example, all strawberries are assumed to be easy to harvest.

This implies that these berries can be successfully harvested by one of the robot arms with mobile base movement. However, if the complexity of harvesting a strawberry is “medium”, it may not be successfully harvested in one attempt. Here, the robot can attempt to harvest it multiple times by adjusting the wrist angle of the arm. In addition, for the strawberry with a “hard” complexity level, which is the most challenging situation where even a human may require using both hands to pick this type of strawberry. Hence, the robot can ignore these types of strawberries and focus on the "easy" and "medium" complexity ones. Generally, the key idea of the berry allocator is constructing a harvest plan with the shared and conflicting resources (body/mobile base). This plan can adjust the action strategy based on the clustering complexity.

6.2.3 Experimental results in the laboratory setting

A simple experiment was demonstrated in the laboratory to verify the feasibility of the strawberry allocator. Herein, a few “easy” strawberries that could be harvested successfully in one attempt were considered. Therefore, the feedback of the perception system resembled that shown in Figure 6.8. This screenshot indicates that the perception system can classify each detected strawberry’s cluster complexity. Furthermore, the 3D coordinates information and picking sequence by which arm were recorded as the harvesting plan.

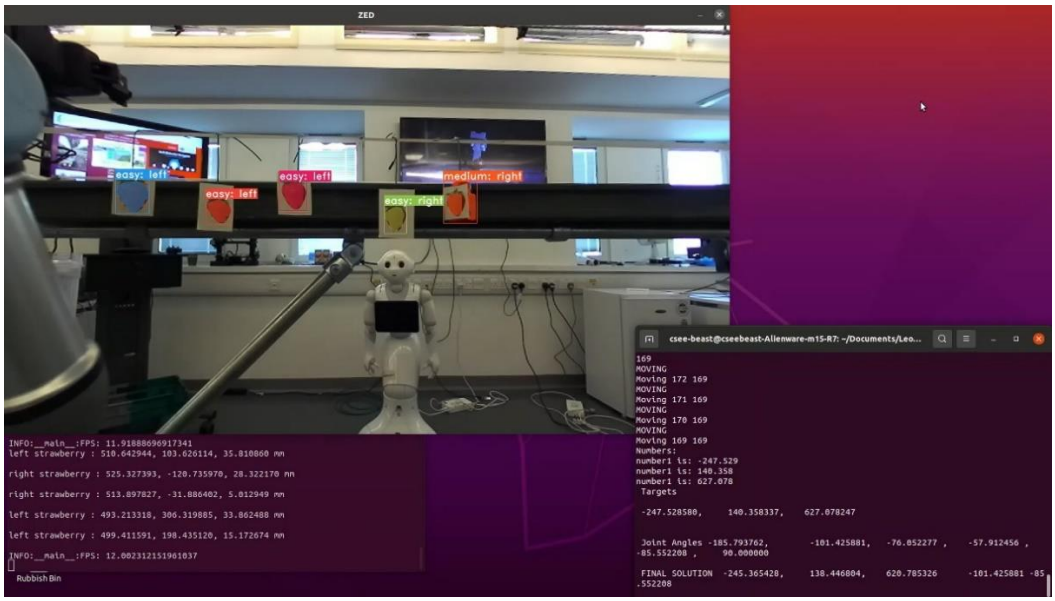


Figure 6. 8 – Perception system display interface in the lab setting.

The first test involved harvesting all strawberries with one arm. As shown in Figure 6.9, the process of the robot harvesting using a single arm was recorded from the robot's perspective. Figure 6.9a shows that the left arm reaches and cuts the first strawberry. Subsequently, the mobile base moves a negligible distance (Figure 6.9b) and harvests the second strawberry (Figure 6.9c). Next, the mobile base moves again for the harvesting the remaining strawberries (Figure 6.9d). In this test, as the robot moved from left to right, the robotic arm needed to pick only the leftmost strawberry one by one.

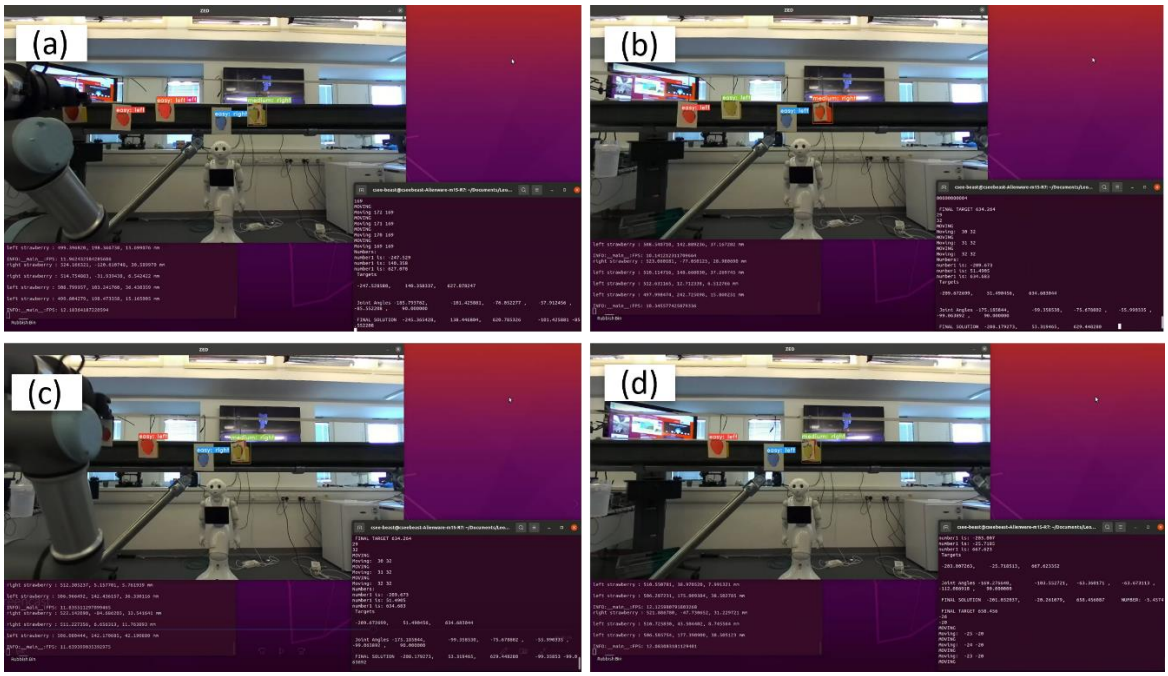


Figure 6. 9 - Demonstration of the single-arm harvesting process in the laboratory.

As the robot moves, a single arm can complete the harvesting task; however, the efficiency of this harvesting is not as high as that of double-arm harvesting. Figure 6.10 demonstrates the harvesting process by the robot using both arms. In this experiment, the mobile base still moved from left to right according to the robot's perspective. Initially, the robot harvested two strawberries located in the left-most working space of each arm (Figure 6.10a). Subsequently, the mobile base moved closer till one strawberry was in front of the left arm (Figure 6.10b), and the two arms harvested the strawberries located in the left-most working space again (Figure 6.10c). Finally, the robot moved and harvested the remaining strawberries (Figure 6.10d). All the laboratory and field demonstrations of this thesis were saved as media files and can be found in the Supplementary Materials of the thesis.

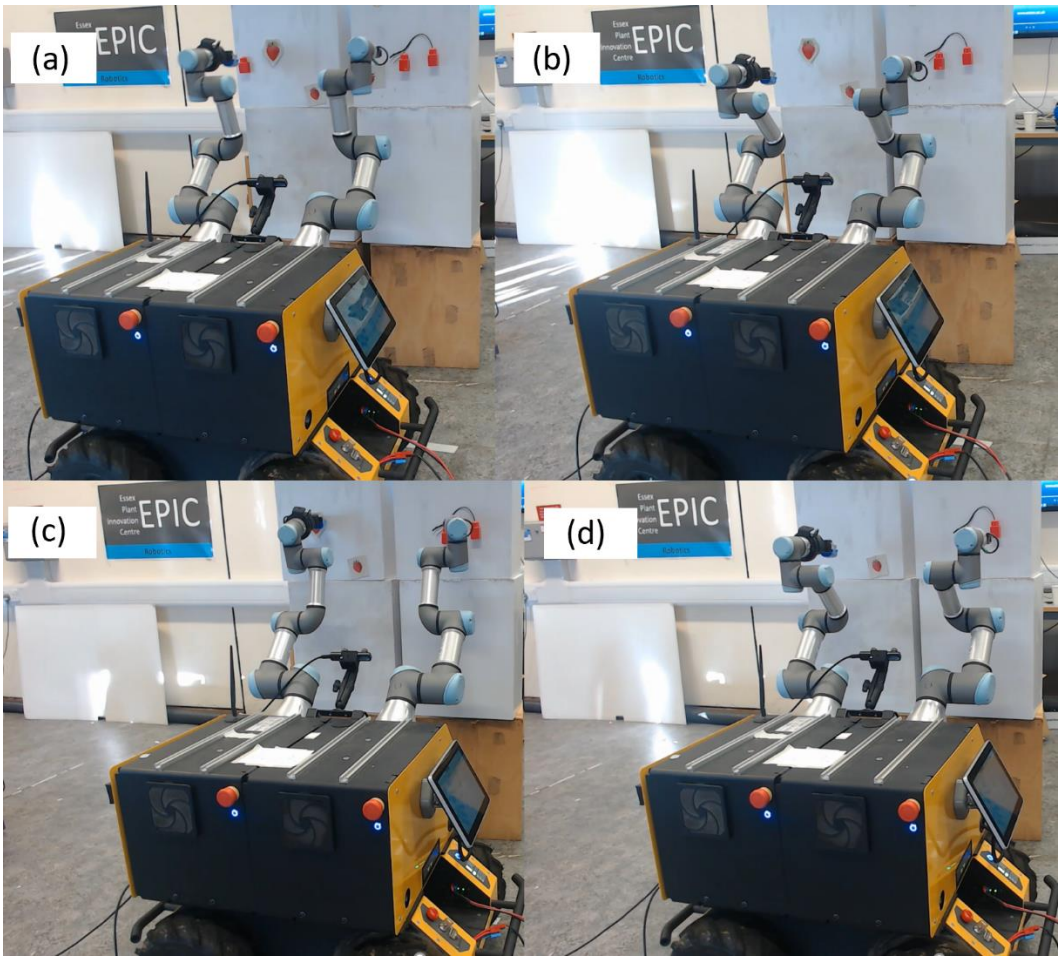


Figure 6. 10 – Demonstration of the dual-arm harvesting process in the laboratory. (a) shows that the robot detects six strawberries and the arms try to pick them from left to right within its own working space; (b) shows that after the first round of picking, the robot moves forward a short distance for the second round of picking (c); (d) shows the robot attempting to pick the remaining strawberries.

6.3 Verifying Robotic Perception–action in Field Application

The field experiments were conducted in the year 2022 in the vertical greenhouse in Tiptree Essex. Unlike the previous round of field experiments (Chapter 5.3), this experiment tested the performance of the integrated system based on the strawberry allocator in farm harvesting. First, the robot based on the integrated system was tested by harvesting "easy," "medium," and "hard" strawberries separately; the harvesting results of a single attempt are recorded in Table 6.1. Evidently from the testing, all easy cases were successfully harvested in a single attempt; however, in two cases, the target

strawberries were dropped after cutting. Although the most of “medium” ones were harvested successfully, in some cases, the robotic arms simultaneously picked ripe and unripe strawberries. However, a few “medium” strawberries and the most of “hard” strawberries that were harvested failed in single attempt harvesting. When the robot was allowed to harvest the “hard” strawberries in two or three attempts, the gripper still cut or damaged some unripe strawberries. Figure 6.11 illustrates the failure cases in the field trials.

Table 6. 1 - Harvesting strawberries success rate with different complexity levels

Complexity level	Success	Failure	Success rate (%)
Easy	21	2	91.3
Medium	19	6	76
Hard	3	11	21.43

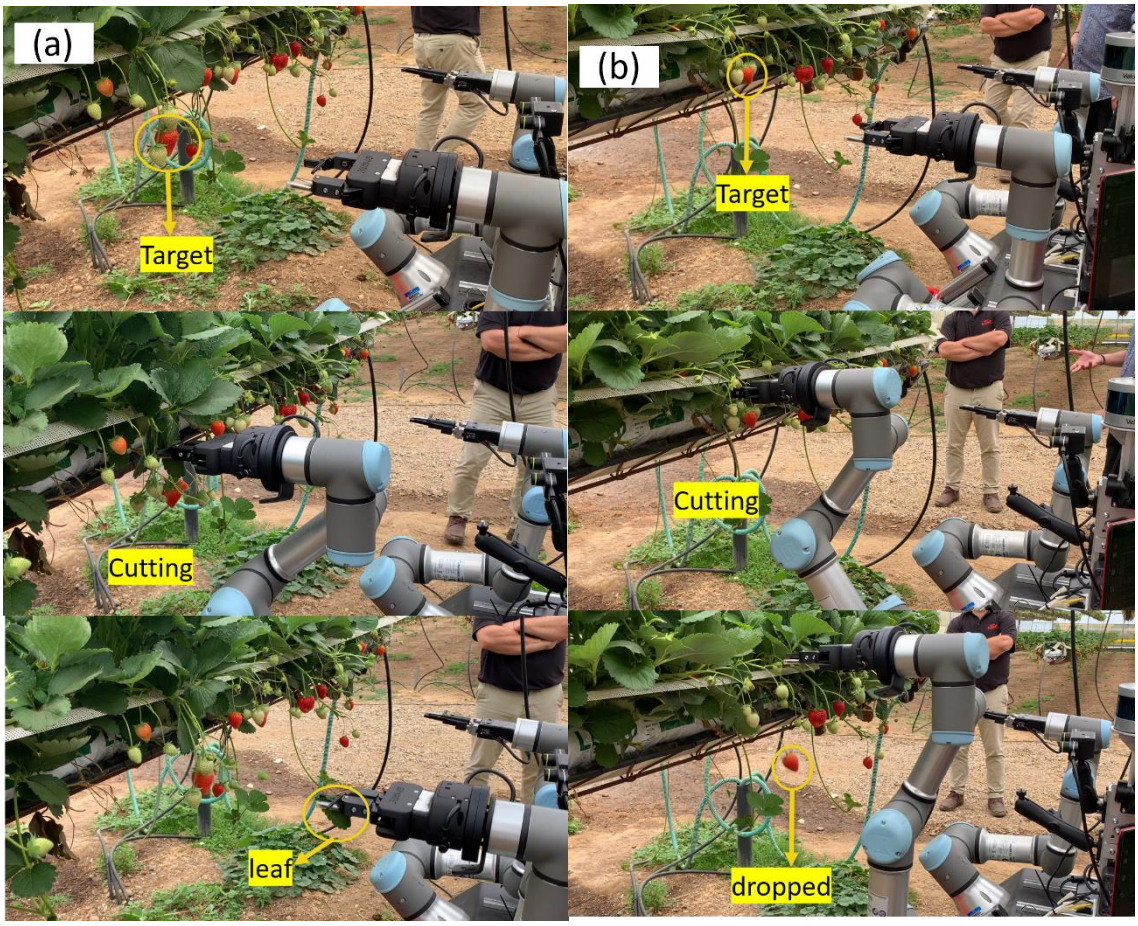


Figure 6. 11 – Failure cases in field test: (a) “Hard” target with obstacles, the gripper only cut the leaf; (b) “Medium” target surrounded by one unripe berry; (c) Target dropped after harvesting.

The robotic arms were not expected to harvest successfully in this challenging environment with plenty of obstacles (i.e., stems, raw strawberries, and leaves), particularly in only a single attempt. Because the perception system can sort out the hard-to-pick strawberries, the robot can ignore the sub-type (“hard”) strawberries and focus on continuously harvesting other strawberries, thus minimising damage to unripe strawberries. However, the “easy” or “medium” strawberries cannot be always harvested successfully in a single attempt. Figure 6.12 shows an example that demonstrates the

robot's ability to improve the harvesting process by moving the mobile base to adjust its position after failing in the first attempt.

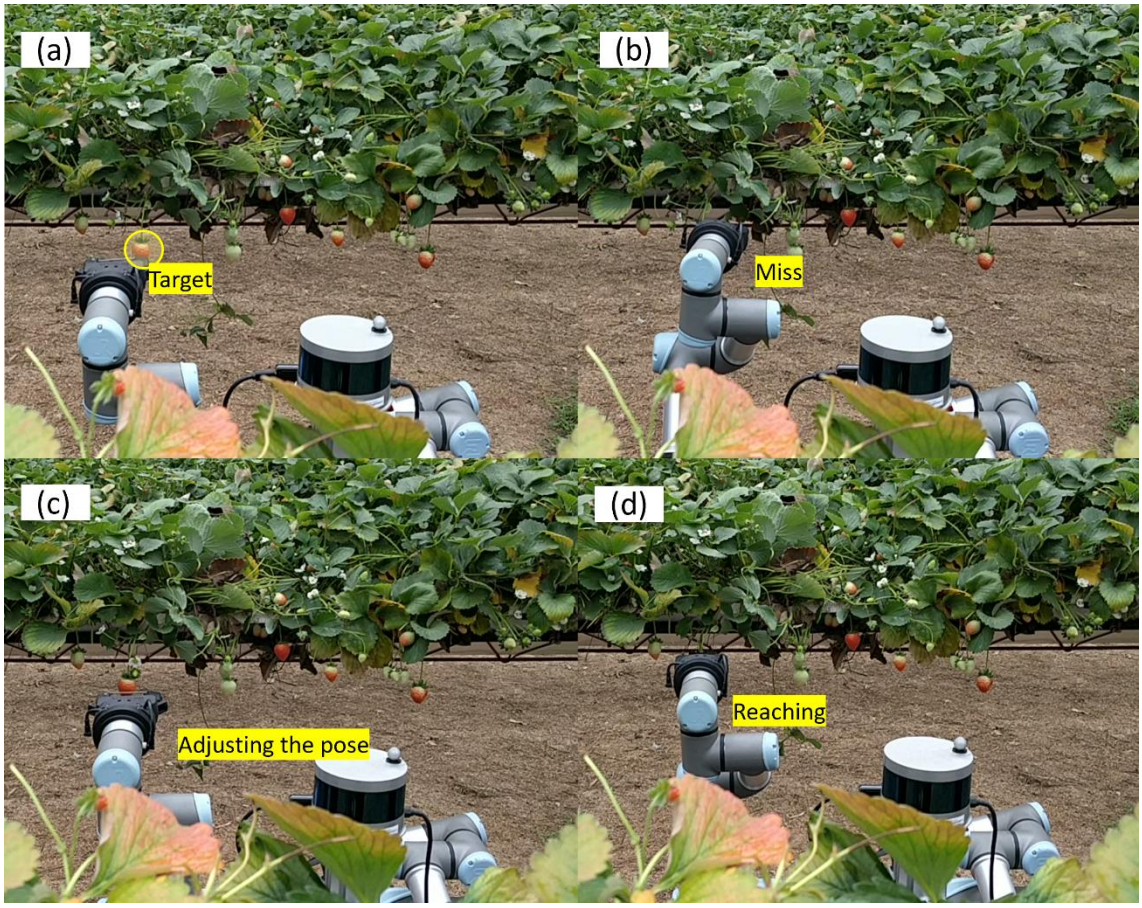


Figure 6. 12 - Cutting a strawberry by adjusting the mobile base. Because the perception system is always updating the coordinates of the strawberry, when the robot attempts to pick but fails, the action system adjusts the pose according to the real-time coordinates of the strawberry and attempts to pick again.

As shown above, although the robot occasionally could not harvest a strawberry with low harvesting complexity in the first attempt, the perception system continued updating the 3D information of the target. Therefore, the location of the target changed slightly. Consequently, the mobile base moved a small distance back and forth to accommodate the change in target pose and attempted to reach again.

In the above field experiments, the proposed robotics system demonstrated good performance in strawberry picking. Currently, the system can perform strawberry harvesting tasks with low cluster complexity. Even if the model ignores strawberries that

are heavily complex to harvest, a human–robot collaboration system can be used for strawberry picking to reduce manual labour.

6.4 Directions and Guidelines for Improvement

Harvesting ripe strawberries in commercial greenhouses using robots presents considerable room for improvement. Existing challenges include analysis of strawberry ripeness, dual-arm collaboration for strawberry harvesting in the cluster, and efficient harvesting with a low damage rate. The proof-of-concept experiments herein provided some research outcomes for adapting the proposed architecture to other fruit. Cluster complexity-based detection and forward action plan can be applied in the perception system to harvest other crops. This can allow harvesters to adopt more targeted action strategies based on the difficulty of harvesting. In this study, the harvesting complexity was classified by the occlusion degree and the dataset was labelled according to the authors' judgment. This may result in classification errors; for example, the perception system occasionally confused some "medium" and "hard" strawberries. In future work, a more quantitative method can be developed to classify the harvesting complexity to assist in data labelling.

Evidently from the field trials, although the damage rate could be reduced using a gripper with a blade to cut the stem, some failure cases were observed. First, a few strawberries were dropped after the gripper cut the stem; this can be avoided by adjusting the size of the blade and gripper. However, the main failure cases were caused by occlusion because of the presence of some leaves, stems, and unripe strawberries; this presents challenges for a single gripper to complete the harvesting. To harvest the strawberry with a high level of complexity, dual-arm collaboration with different grippers can be used. Essentially, the action system can be extended to control two hands for harvesting strawberries in the cluster, similar to humans.

Overall, the field-evaluated robotics platform can aid the further development of agricultural robotics systems for other crops.

6.5 Conclusions and Open Questions

This chapter presented a dual-arm robotics system that demonstrated an automated approach for harvesting strawberries. The proposed system comprised a cluster complexity-based perception and a berry allocator harvesting strategy was integrated with the action system. The efficiency of the system was verified in a vertical growing system in England.

The perception system aimed to determine the harvesting complexity of ripe strawberries; this can help the harvesters adopt different action strategies to handle strawberries with different complexity levels. Another contribution of the system was proposing a berry allocator to help the dual-arm robot harvest strawberries reasonably with shared and conflicting resources. Furthermore, the action system was developed based on the PMP for stem-directed cutting with a 3D-printed gripper, which avoided directly touching the strawberry to reduce the damage rate. The field experimental results revealed that the proposed architecture can simultaneously pick low complexity level strawberries with two hands and avoid damaging the high complexity level strawberries. The concepts allow some future extensions and further work, as follows.

1) *Harvesting high complexity level of strawberries*. Although using the robotics arm to precisely harvest ripe strawberries with considerable occlusion is challenging, this chapter introduced the idea of constructing harvesting strategies based on the clustering complexity. Thus, dual-arm collaboration with different grippers has considerable potential for future research.

2) *Configurability to other crops*. In addition to strawberry harvesting, other fruit harvesting robots also face cluster complex problems. The proposed approach can be considered for soft fruit and various cross-industry applications.

3) *Diagnosing diseases by combining mobile robots and drones*. This study focused on strawberry harvesting; however, in the fruit industry, pests and diseases severely affect the yield of fruits. Thus, diagnosis and pest/disease treatment using innovations in mobile robots and drones can increase fruits and vegetable production.

Chapter 7

General Conclusions and Future Work

This chapter summarises the contributions of this thesis and closes with future extensions and possible research directions.

7.1 Summary and Extension

This study presented an adaptive, biomimetic, and configurable robot for smart farms, with a focus on the perception–action system for cereal/soft fruit phenotyping/identification and harvesting tasks. First, the application of machine vision phenotyping of the wheat plant was investigated. Further, recognition technology and robotic control for harvesting strawberries, a common soft fruit, were investigated in a commercial greenhouse. The results are as follows.

First, an adaptive *k*-means algorithm with dynamic perspectives was developed to separate the wheat spikes, remove stems, and obtain spikes. To realise field application, the method randomly selected some areas as sample areas and called the algorithm to calculate the average spike size. Furthermore, the algorithm segmented all the spikes as thousands of small segments and used cuboids to fit each segment and estimate the total volume of all spikes. However, the fitting errors increased when this method was extended to wheat spikes that were curved and overlapping. Therefore, future work is expected to optimise the algorithm further to handle the environment where the wheat spike is arched. Although the proposed method is based on the classical clustering algorithm, which can omit the training process with fewer computing resources, it does

not imply that the deep-learning models cannot handle this problem. Contrary, further work can develop a robust deep-learning model to address the issues opened up by the current work.

Second, to detect ripe strawberries, a cGAN model was trained on synthetically generated data, which included various lighting conditions and occlusions as observed in real-world conditions. This approach alleviated the difficulty in collecting and labelling data during the pandemic. However, after one round of field experiments, the distribution of obstacles was found to affect the difficulty in strawberry harvesting. Therefore, a YOLACT-based fruit cluster complexity model was proposed to guide the robot to determine whether a strawberry was easy to harvest.

Further work can attempt to identify overripe strawberries, which is crucial for commercial farms. Strawberries are susceptible to numerous diseases in the growing chain, and these can be divided primarily into two categories: infectious and physiological diseases. These diseases can affect the strawberry leaves, fruits, or flowers [113]. Deep-learning models for detecting strawberry diseases have been developed recently [114]. Therefore, in future, efforts to layer a dataset of healthy fruits with pictures of "rot" of similar shape and size, and then apply occlusion filters to these images to synthesise "rotten" fruits pictures can be made. This new data can be used to simulate the presence of diseases on strawberries by randomly placing "rot" occlusion images (which are other darker-coloured objects and fruits) and retraining the perceptual system, which justifies the use of flexible hypothetical data for real-world situations. The figure 7.1 illustrates only one type of rot (strawberries typically develop black or white regions when they rot or get infected with bacteria).

Furthermore, future work can aim to automate scouting, diagnosis, and pest/disease treatment using innovations in swarm robotics, drones, and AI that are generalisable for

producing more expansive fruit. Further directions for future works can also be considered. Given the previous studies on wheat plants, phenotypic analysis, fruit counting, and weight estimation can be extended to soft fruit for yield analysis.

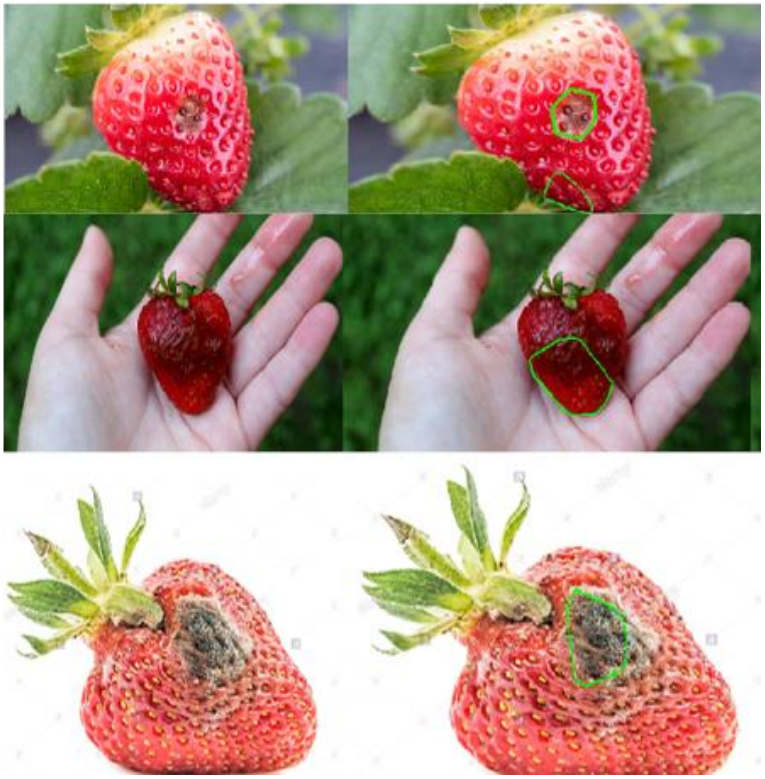


Figure 7. 1 - Example of the rotten strawberry recognition. The left side presents some original images, and the right side shows the corresponding post-detection images.

Third, to move the end-effector accurately toward the fruit, an action system based on a neural network implementation of the PMP was developed for the robotic arms. The results illustrated goal-directed reaching and exhibited a mean error of less than 3 mm in the laboratory. Further, the action system was integrated with the perception system for field-tested in a state-of-the-art vertical growing system. According to the field testing, when the robot utilised the shared and conflicting resources (arms/mobile base) properly, the harvesting efficiency and success rate increased. Therefore, a strawberry allocator was proposed to construct a forward harvesting plan that can help the robot determine the harvesting sequence of all detected strawberries using one arm. The second round of field experiments verified the integrated action–perception system; this implies that the robot

not only knows the harvesting complexity of strawberries but also can harvest the strawberries by coordinating the arms and mobile base.

7.2 Possible Research Direction

Future work can further exploit the dual-arm coordination for strawberry picking with high harvesting complexity level. For example, removing the obstacles (i.e., stems, leaves, and unripe strawberries) with one hand and picking the target strawberry with the other hand. This also requires developing a more dexterous gripper that does not damage any strawberries in the harvesting process. Additionally, the current system estimates the stem position based on the bounding box of the detected strawberry. It can be further improved using the 6D pose estimation method. In addition to robots enabling autonomous harvesting on farms, the future research direction may be summarised as follow.

Scouting.

The flowers, leaves, and flesh of the fruit are susceptible to disease during growth. These features of diseases are diverse and complex. The number of fruits counted and the analysis of diseases on the seed, leaves, flowers, and flesh would therefore be a branch of future research. In addition, fruit maturity, quantitative, and phenotypic analyses are useful for yield prediction.

Monitoring.

In smart agriculture, intelligent monitoring and control systems have become a research highlight. For example, the control of temperature, humidity, energy consumption, and the oxygen content of nutrient solutions in greenhouses can effectively improve crop yields. This monitoring system could involve the internet of things, sensor fusion, and intelligent decision-making, among many other technologies that could be the subject of further research.

Transport and packaging.

In an integrated intelligent farming plant, the handling and packaging of fruit and vegetables can also be carried out by robots and automatic sorting systems. From the literature [115], it is clear that although researchers have focused on individual aspects of processing and packaging, there is a need for a more holistic approach to system analysis while understanding the scope of the entire operations. Therefore, a scheduling system can be built to ensure that multiple robots can be operated and transported in an orderly and efficient manner.

Safety and Collaboration.

Safety is a very important factor for human-robot collaboration. It includes collision avoidance between robotic arms and avoidance between robots and operators. These can be implemented in software by developing appropriate algorithms for avoidance, but also require the development of appropriate hardware such as sensors to improve safety.

The general research direction can focus on efficiently deploying intelligent mobile robots, manipulators, and unmanned aerial vehicles collaboratively for automating complex and labour-intensive tasks to ensure yield and quality of crop production. The figure below illustrates one example vision of future work: developing a fleet of adaptive and cost-effective collaborative robots operating in a smart farm and adaptively configured to automate/learn a range of harvesting tasks (i.e., weeding, picking, packaging, scouting, and crop intelligence/protection). Modular hardware/software architecture can allow new functionality to be added cumulatively. Further, the focus is also on novel workflows for human–robot collaboration, safety/trust, explainability, and intuitive user interfaces for robotics fleet operation/visualisation of the farm data.

Figure 7. 2 - Vision for future work. In this example, cost-effectiveness can be achieved using low-cost robotic arms with better payload and repeat accuracy, low-cost/low-power embedded processing hardware, and 3D-printed end-effectors/tools.

Finally, this study expects that advanced robotics, AI, and computer vision will bring much-needed versatility to smart farming. Smart farms are revolutionising farming and food production, and will significantly transform the food we eat and how we produce it.

Supplementary Material

The media files demonstrating the findings of this study's perception and action system is availabel：

https://youtube.com/playlist?list=PL2ukSjWhuNP9balDtgh_U51ds1oXPjn0d

In the meantime, the corresponding media file can be found in the author's published literature:

https://doi.org/10.1007/s11119-023-10000-4

Bibliography

[1] R. Sparrow and M. Howard, “Robots in agriculture: prospects, impacts, ethics, and policy,” *Precis. Agric.*, vol. 22, no. 3, pp. 818–833, 2021, doi: 10.1007/s11119-020-09757-9.

[2] “British Summer Fruits - Fruit Focus 2023.” https://www.fruitfocus.co.uk/partners/british-summer-fruits (accessed May 15, 2023).

[3] “Food - Worldwide | Statista Market Forecast.” https://www.statista.com/outlook/cmo/food/worldwide (accessed Apr. 25, 2022).

[4] S. Rotz *et al.*, “Automated pastures and the digital divide: How agricultural technologies are shaping labour and rural communities,” *J. Rural Stud.*, vol. 68, pp. 112–122, 2019, doi: https://doi.org/10.1016/j.jrurstud.2019.01.023.

[5] J. Houghton, “Global warming,” *Reports Prog. Phys.*, vol. 68, no. 6, p. 1343, 2005, doi: 10.1088/0034-4885/68/6/R02.

[6] M. J. Salinger, “Climate Variability and Change: Past, Present and Future --- an Overview,” in *Increasing Climate Variability and Change: Reducing the Vulnerability of Agriculture and Forestry*, J. Salinger, M. V. K. Sivakumar, and R. P. Motha, Eds. Dordrecht: Springer Netherlands, 2005, pp. 9–29.

[7] M. Ofori and O. El-Gayar, “Drivers and challenges of precision agriculture: a social media perspective,” *Precis. Agric.*, vol. 22, no. 3, pp. 1019–1044, 2021, doi: 10.1007/s11119-020-09760-0.

[8] D.-G. for C. (European Commission), “The EU in 2021,” 2022. doi: NA-AD-22-001-EN-N.

[9] W. Boedeker, M. Watts, P. Clausing, and E. Marquez, “The global distribution of acute unintentional pesticide poisoning: estimations based on a systematic review,” *BMC Public Health*, vol. 20, no. 1, p. 1875, 2020, doi: 10.1186/s12889-020-09939-0.

[10] D. Rakita, B. Mutlu, M. Gleicher, and L. M. Hiatt, “Shared control-based bimanual robot manipulation,” *Sci. Robot.*, vol. 4, no. 30, p. eaaw0955, 2019, doi: 10.1126/scirobotics.aaw0955.

[11] Y. Zhao, L. Gong, Y. Huang, and C. Liu, “A review of key techniques of vision-based control for harvesting robot,” *Comput. Electron. Agric.*, vol. 127, pp. 311–323, 2016, doi: 10.1016/j.compag.2016.06.022.

[12] Y. Xu, K. Imou, Y. Kaizu, and K. Saga, “Two-stage approach for detecting slightly overlapping strawberries using HOG descriptor,” *Biosyst. Eng.*, vol. 115, no. 2, pp. 144–153, 2013, doi: https://doi.org/10.1016/j.biosystemseng.2013.03.011.

[13] Y. Bai, S. Mao, J. Zhou, and B. Zhang, “Clustered tomato detection and picking point location using machine learning-aided image analysis for automatic robotic harvesting,” *Precis. Agric.*, vol. 24, no. 2, pp. 727–743, 2023, doi: 10.1007/s11119-022-09972-6.

[14] Z. Su, F. Wang, H. Xiao, H. Yu, and S. Dong, “A fault diagnosis model based on singular value manifold features, optimized SVMs and multi-sensor information fusion,” *Meas. Sci. Technol.*, vol. 31, no. 9, p. 95002, Jun. 2020, doi: 10.1088/1361-6501/ab842f.

[15] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, “Object Detection with Deep Learning: A Review,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, 2019, doi: 10.1109/TNNLS.2018.2876865.

[16] T. M. Navamani, “Chapter 7 - Efficient Deep Learning Approaches for Health Informatics,” in *Deep Learning and Parallel Computing Environment for Bioengineering Systems*, A. K. Sangaiah, Ed. Academic Press, 2019, pp. 123–137.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, 2020, doi: 10.1109/TPAMI.2018.2844175.

[18] Y. Yu, K. Zhang, L. Yang, and D. Zhang, “Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN,” *Comput. Electron. Agric.*, vol. 163, 2019, doi: 10.1016/j.compag.2019.06.001.

[19] T. T. Santos, L. L. de Souza, A. A. dos Santos, and S. Avila, “Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association,” *Comput. Electron. Agric.*, vol. 170, p. 105247, 2020, doi: https://doi.org/10.1016/j.compag.2020.105247.

[20] Y. Ge, Y. Xiong, G. L. Tenorio, and P. J. From, “Fruit Localization and Environment Perception for Strawberry Harvesting Robots,” *IEEE Access*, vol. 7, pp. 147642–147652, 2019, doi: 10.1109/ACCESS.2019.2946369.

[21] H. Altaheri, M. Alsulaiman, and G. Muhammad, “Date Fruit Classification for Robotic Harvesting in a Natural Environment Using Deep Learning,” *IEEE Access*, vol. 7, pp. 117115–117133, 2019, doi: 10.1109/access.2019.2936536.

[22] S. Birrell, J. Hughes, J. Y. Cai, and F. Iida, “A field-tested robotic harvesting system for iceberg lettuce,” *J. F. Robot.*, vol. 37, no. 2, pp. 225– 245, 2020, doi: 10.1002/rob.21888.

[23] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” *arXiv*, 2018, doi: 10.48550/ARXIV.1804.02767.

[24] L. Li, Q. Zhang, and D. Huang, “A review of imaging techniques for plant phenotyping,” *Sensors (Switzerland)*, vol. 14, no. 11, pp. 20078–20111, 2014, doi: 10.3390/s141120078.

[25] A. Pask, J. Pietragalla, and D. Mullan, *Physiological Breeding II: A Field Guide to Wheat Phenotyping*. 2012.

[26] D. Deery, J. Jimenez-Berni, H. Jones, X. Sirault, and R. Furbank, “Proximal remote sensing buggies and potential applications for field-based phenotyping,”

Agronomy, vol. 4, no. 3, pp. 349–379, 2014, doi: 10.3390/agronomy4030349.

[27] J. A. Fernandez-Gallego, S. C. Kefauver, N. A. Gutiérrez, M. T. Nieto-Taladriz, and J. L. Araus, "Wheat ear counting in-field conditions: High throughput and low-cost approach using RGB images," *Plant Methods*, vol. 14, 2018, doi: 10.1186/s13007-018-0289-4.

[28] P. Sadeghi-Tehran, N. Virlet, E. M. Ampe, P. Reyns, and M. J. Hawkesford, "DeepCount: In-Field Automatic Quantification of Wheat Spikes Using Simple Linear Iterative Clustering and Deep Convolutional Neural Networks," *Front. Plant Sci.*, vol. 10, 2019, doi: 10.3389/fpls.2019.01176.

[29] C. Tan *et al.*, "Rapid Recognition of Field-Grown Wheat Spikes Based on a Superpixel Segmentation Algorithm Using Digital Images," *Front. Plant Sci.*, vol. 11, p. 259, 2020, doi: 10.3389/fpls.2020.00259.

[30] S. Dandrifosse, E. Ennadifi, A. Carlier, B. Gosselin, B. Dumont, and B. Mercatoris, "Deep learning for wheat ear segmentation and ear density measurement: From heading to maturity," *Comput. Electron. Agric.*, vol. 199, p. 107161, 2022, doi: https://doi.org/10.1016/j.compag.2022.107161.

[31] I. Mohamed and R. Dudley, "Comparison of 3D Imaging Technologies for Wheat Phenotyping," *{IOP} Conf. Ser. Earth Environ. Sci.*, vol. 275, p. 12002, May 2019, doi: 10.1088/1755-1315/275/1/012002.

[32] H. Su *et al.*, "SPLATNet: Sparse Lattice Networks for Point Cloud Processing," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2530–2539, doi: 10.1109/CVPR.2018.00268.

[33] L. Li, M. Sung, A. Dubrovina, L. Yi, and L. J. Guibas, "Supervised fitting of geometric primitives to 3D point clouds," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2647–2655, doi: 10.1109/CVPR.2019.00276.

[34] K. Velumani, S. Oude Elberink, M. Y. Yang, and F. Baret, "Wheat Ear Detection in Plots by Segmenting Mobile Laser Scanner Data," in *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2017, pp. 149–156, doi: 10.5194/isprs-annals-IV-2-W4-149-2017.

[35] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[36] A. Thompson, V. Livina, P. Harris, I. Mohamed, and R. Dudley, "Model-based algorithms for phenotyping from 3D imaging of dense wheat crops," in *2019 IEEE International Workshop on Metrology for Agriculture and Forestry, MetroAgriFor 2019 - Proceedings*, 2019, pp. 95–99, doi: 10.1109/MetroAgriFor.2019.8909214.

[37] H. Zhou, X. Wang, W. Au, H. Kang, and C. Chen, "Intelligent robots for fruit harvesting: recent developments and future challenges," *Precis. Agric.*, vol. 23, no. 5, pp. 1856–1907, 2022, doi: 10.1007/s11119-022-09913-3.

[38] T. Zhang, Z. Huang, W. You, J. Lin, X. Tang, and H. Huang, "An Autonomous

Fruit and Vegetable Harvester with a Low-Cost Gripper Using a 3D Sensor," *Sensors*, vol. 20, no. 1, 2020, doi: 10.3390/s20010093.

[39] F. Dimeas, D. V Sako, V. C. Moulianitis, and N. A. Aspragathos, "Design and fuzzy control of a robotic gripper for efficient strawberry harvesting," *Robotica*, vol. 33, no. 5, pp. 1085–1098, 2015, doi: 10.1017/S0263574714001155.

[40] H. Yaguchi, K. Nagahama, T. Hasegawa, and M. Inaba, "Development of an autonomous tomato harvesting robot with rotational plucking gripper," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 652–657, doi: 10.1109/IROS.2016.7759122.

[41] A. Silwal, J. R. Davidson, M. Karkee, C. Mo, Q. Zhang, and K. Lewis, "Design, integration, and field evaluation of a robotic apple harvester," *J. F. Robot.*, vol. 34, no. 6, pp. 1140–1159, 2017, doi: https://doi.org/10.1002/rob.21715.

[42] K. Tanigaki, T. Fujiura, A. Akase, and J. Imagawa, "Cherry-harvesting robot," *Comput. Electron. Agric.*, vol. 63, no. 1, pp. 65–72, 2008, doi: https://doi.org/10.1016/j.compag.2008.01.018.

[43] D. DeMers and K. Kreutz-Delgado, "4 - Inverse Kinematics of Dextrous Manipulators," in *Neural Systems for Robotics*, O. Omidvar and P. van der Smagt, Eds. Boston: Academic Press, 1997, pp. 75–116.

[44] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *J. Neurosci.*, vol. 5, no. 7, pp. 1688–1703, 1985, doi: 10.1523/JNEUROSCI.05-07-01688.1985.

[45] P. Morasso, "Spatial control of arm movements," *Exp. Brain Res.*, vol. 42, no. 2, pp. 223–227, 1981, doi: 10.1007/BF00236911.

[46] Y. Uno, M. Kawato, and R. Suzuki, "Formation and control of optimal trajectory in human multijoint arm movement," *Biol. Cybern.*, vol. 61, no. 2, pp. 89–101, 1989, doi: 10.1007/BF00204593.

[47] J. B. Dingwell, C. D. Mah, and F. A. Mussa-Ivaldi, "Experimentally Confirmed Mathematical Model for Human Control of a Non-Rigid Object," *J. Neurophysiol.*, vol. 91, no. 3, pp. 1158–1170, 2004, doi: 10.1152/jn.00704.2003.

[48] S. Ben-Itzhak and A. Karniel, "Minimum Acceleration Criterion with Constraints Implies Bang-Bang Control as an Underlying Principle for Optimal Trajectories of Arm Reaching Movements," *Neural Comput.*, vol. 20, no. 3, pp. 779–812, 2008, doi: 10.1162/neco.2007.12-05-077.

[49] R. Shadmehr, M. A. Smith, and J. W. Krakauer, "Error Correction, Sensory Prediction, and Adaptation in Motor Control," *Annu. Rev. Neurosci.*, vol. 33, no. 1, pp. 89–108, 2010, doi: 10.1146/annurev-neuro-060909-153135.

[50] D. Liu and E. Todorov, "Evidence for the flexible sensorimotor strategies predicted by optimal feedback control," *J. Neurosci.*, vol. 27, no. 35, p. 9354—9368, Aug. 2007, doi: 10.1523/jneurosci.1110-06.2007.

[51] J. J. Kutch, A. D. Kuo, A. M. Bloch, and W. Z. Rymer, "Endpoint Force Fluctuations Reveal Flexible Rather Than Synergistic Patterns of Muscle Cooperation," *J. Neurophysiol.*, vol. 100, no. 5, pp. 2455–2471, 2008, doi: 10.1152/jn.90274.2008.

[52] P. Beeson and B. Ames, “TRAC-IK: An open-source library for improved solving of generic inverse kinematics,” in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 928–935, doi: 10.1109/HUMANOIDS.2015.7363472.

[53] V. Mohan and P. Morasso, “Passive motion paradigm: An alternative to optimal control,” *Front. Neurorobot.*, vol. 5, 2011, doi: 10.3389/fnbot.2011.00004.

[54] S. H. Scott, “Optimal feedback control and the neural basis of volitional motor control,” *Nat. Rev. Neurosci.*, vol. 5, no. 7, pp. 532–545, 2004, doi: 10.1038/nrn1427.

[55] E. Bizzi, A. Polit, and P. Morasso, “Mechanisms underlying achievement of final head position,” *J. Neurophysiol.*, vol. 39, no. 2, pp. 435–444, 1976, doi: 10.1152/jn.1976.39.2.435.

[56] E. Bizzi, N. Hogan, F. A. Mussa-Ivaldi, and S. Giszter, “Does the nervous system use equilibrium-point control to guide single and multiple joint movements?,” *Behav. Brain Sci.*, vol. 15, no. 4, pp. 603–613, 1992, doi: 10.1017/S0140525X00072538.

[57] A. G. Feldman and M. F. Levin, “The origin and use of positional frames of reference in motor control,” *Behav. Brain Sci.*, vol. 18, no. 4, pp. 723–744, 1995, doi: 10.1017/S0140525X0004070X.

[58] J. Roh, V. C. K. Cheung, and E. Bizzi, “Modules in the brain stem and spinal cord underlying motor behaviors,” *J. Neurophysiol.*, vol. 106, no. 3, pp. 1363–1378, 2011, doi: 10.1152/jn.00842.2010.

[59] M. Berniker, A. Jarc, E. Bizzi, and M. C. Tresch, “Simplified and effective motor control based on muscle synergies to exploit musculoskeletal dynamics,” *Proc. Natl. Acad. Sci.*, vol. 106, no. 18, pp. 7601–7606, 2009, doi: 10.1073/pnas.0901512106.

[60] E. Bizzi, F. A. Mussa-Ivaldi, and S. Giszter, “Computations Underlying the Execution of Movement: A Biological Perspective,” *Science (80-.).*, vol. 253, no. 5017, pp. 287–291, 1991, doi: 10.1126/science.1857964.

[61] A. d’Avella and E. Bizzi, “Shared and specific muscle synergies in natural motor behaviors,” *Proc. Natl. Acad. Sci.*, vol. 102, no. 8, pp. 3076–3081, 2005, doi: 10.1073/pnas.0500199102.

[62] F. A. Mussa-Ivaldi and E. Bizzi, “Motor learning through the combination of primitives,” *Philos. Trans. R. Soc. B Biol. Sci.*, vol. 355, no. 1404, pp. 1755 – 1769, 2000, doi: 10.1098/rstb.2000.0733.

[63] N. Hogan, “Impedance Control: An Approach to Manipulation: Part I—Theory,” *J. Dyn. Syst. Meas. Control*, vol. 107, no. 1, pp. 1–7, 1985, doi: 10.1115/1.3140702.

[64] F. A. M. Ivaldi, P. Morasso, and R. Zaccaria, “Kinematic networks,” *Biol. Cybern.*, vol. 60, no. 1, pp. 1–16, 1988, doi: 10.1007/BF00205967.

[65] V. Mohan, P. Morasso, G. Metta, and G. Sandini, “A biomimetic, force-field based computational model for motion planning and bimanual coordination in humanoid robots,” *Auton. Robots*, vol. 27, no. 3, p. 291, 2009, doi:

10.1007/s10514-009-9127-x.

[66] V. Mohan, A. Bhat, and P. Morasso, “Muscleless motor synergies and actions without movements: From motor neuroscience to cognitive robotics,” *Phys. Life Rev.*, vol. 30, pp. 89–111, 2018, doi: 10.1016/j.plrev.2018.04.005.

[67] P. Morasso, M. Casadio, V. Mohan, and J. Zenzeri, “A neural mechanism of synergy formation for whole body reaching,” *Biol. Cybern.*, vol. 102, no. 1, pp. 45–55, 2010, doi: 10.1007/s00422-009-0349-y.

[68] V. Mohan, P. Morasso, J. Zenzeri, G. Metta, V. S. Chakravarthy, and G. Sandini, “Teaching a humanoid robot to draw ‘Shapes,’” *Auton. Robots*, vol. 31, no. 1, pp. 21–53, 2011, doi: 10.1007/s10514-011-9229-0.

[69] G. Sandini, V. Mohan, A. Sciutti, and P. Morasso, “Social Cognition for Human-Robot Symbiosis-Challenges and Building Blocks.,” *Front. Neurorobot.*, vol. 12, p. 34, 2018, doi: 10.3389/fnbot.2018.00034.

[70] X. Ling, Y. Zhao, L. Gong, C. Liu, and T. Wang, “Dual-arm cooperation and implementing for robotic harvesting tomato using binocular vision,” *Rob. Auton. Syst.*, vol. 114, pp. 134–143, 2019, doi: 10.1016/j.robot.2019.01.019.

[71] J. Rong, P. Wang, T. Wang, L. Hu, and T. Yuan, “Fruit pose recognition and directional orderly grasping strategies for tomato harvesting robots,” *Comput. Electron. Agric.*, vol. 202, p. 107430, 2022, doi: https://doi.org/10.1016/j.compag.2022.107430.

[72] J. Kim, H. Pyo, I. Jang, J. Kang, B. Ju, and K. Ko, “Tomato harvesting robotic system based on Deep-ToMaToS: Deep learning network using transformation loss for 6D pose estimation of maturity classified tomatoes with side-stem,” *Comput. Electron. Agric.*, vol. 201, p. 107300, 2022, doi: https://doi.org/10.1016/j.compag.2022.107300.

[73] Y. Xiong, C. Peng, L. Grimstad, P. J. From, and V. Isler, “Development and field evaluation of a strawberry harvesting robot with a cable-driven gripper,” *Comput. Electron. Agric.*, vol. 157, pp. 392–402, 2019, doi: https://doi.org/10.1016/j.compag.2019.01.009.

[74] Y. Xiong, Y. Ge, L. Grimstad, and P. J. From, “An autonomous strawberry-harvesting robot: Design, development, integration, and field evaluation,” *J. F. Robot.*, vol. 37, no. 2, pp. 202– 224, 2020, doi: 10.1002/rob.21889.

[75] Y. Ge, Y. Xiong, and P. J. From, “Three-dimensional location methods for the vision system of strawberry-harvesting robots: development and comparison,” *Precis. Agric.*, vol. 24, no. 2, pp. 764–782, 2023, doi: 10.1007/s11119-022-09974-4.

[76] C. Peng, S. Vougioukas, D. Slaughter, Z. Fei, and R. Arikapudi, “A strawberry harvest-aiding system with crop-transport collaborative robots: Design, development, and field evaluation,” *J. F. Robot.*, vol. 39, no. 8, pp. 1231– 1257, 2022, doi: https://doi.org/10.1002/rob.22106.

[77] C. W. Bac, J. Hemming, and E. J. van Henten, “Robust pixel-based classification of obstacles for robotic harvesting of sweet-pepper,” *Comput. Electron. Agric.*, vol. 96, pp. 148–162, 2013, doi: https://doi.org/10.1016/j.compag.2013.05.004.

[78] B. Arad *et al.*, “Development of a sweet pepper harvesting robot,” *J. F. Robot.*, vol. 37, no. 6, pp. 1027–1039, 2020, doi: 10.1002/rob.21937.

[79] Z. Ning *et al.*, “Recognition of sweet peppers and planning the robotic picking sequence in high-density orchards,” *Comput. Electron. Agric.*, vol. 196, p. 106878, 2022, doi: https://doi.org/10.1016/j.compag.2022.106878.

[80] Z. Wu, R. Yang, F. Gao, W. Wang, L. Fu, and R. Li, “Segmentation of abnormal leaves of hydroponic lettuce based on DeepLabV3+ for robotic sorting,” *Comput. Electron. Agric.*, vol. 190, p. 106443, 2021, doi: https://doi.org/10.1016/j.compag.2021.106443.

[81] Y. Park, H.-J. Kim, and H. Il Son, “Novel attitude control of Korean cabbage harvester using backstepping control,” *Precis. Agric.*, vol. 24, no. 2, pp. 744–763, 2023, doi: 10.1007/s11119-022-09973-5.

[82] A. G. Eguíluz, I. Rañó, S. A. Coleman, and T. M. McGinnity, “Reliable object handover through tactile force sensing and effort control in the Shadow Robot hand,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 372–377, doi: 10.1109/ICRA.2017.7989048.

[83] S. Fountas, N. Mylonas, I. Malounas, E. Rodias, C. Hellmann Santos, and E. Pekkeriet, “Agricultural Robotics for Field Operations,” *Sensors*, vol. 20, no. 9, 2020, doi: 10.3390/s20092672.

[84] J. G. A. Barbedo, C. S. Tibola, and J. M. C. Fernandes, “Detecting Fusarium head blight in wheat kernels using hyperspectral imaging,” *Biosyst. Eng.*, vol. 131, pp. 65–76, 2015, doi: https://doi.org/10.1016/j.biosystemseng.2015.01.003.

[85] W. Lai, M. Zhou, F. Hu, K. Bian, and Q. Song, “A New DBSCAN Parameters Determination Method Based on Improved MVO,” *IEEE Access*, vol. 7, pp. 104085–104095, 2019, doi: 10.1109/ACCESS.2019.2931334.

[86] R. Schnabel, R. Wahl, and R. Klein, “Efficient RANSAC for point-cloud shape detection,” *Comput. Graph. Forum*, vol. 26, pp. 214–226, 2007, doi: 10.1111/j.1467-8659.2007.01016.x.

[87] D. Cai, “Litekmeans: the fastest matlab implementation of kmeans,” *Software available at: http://www. zjucadcg. cn/dengcai/Data/Clustering. html*, 2011. http://www.cad.zju.edu.cn/home/dengcai/Data/code/litekmeans.m.

[88] S. Xia *et al.*, “A Fast Adaptive k-means with No Bounds,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 87–89, 2020, doi: 10.1109/tpami.2020.3008694.

[89] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, “A survey of modern deep learning based object detection models,” *Digit. Signal Process.*, vol. 126, p. 103514, 2022, doi: https://doi.org/10.1016/j.dsp.2022.103514.

[90] M. Rahnemoonfar and C. Sheppard, “Deep count: Fruit counting based on deep simulated learning,” *Sensors (Switzerland)*, vol. 17, no. 4, p. 905, 2017, doi: 10.3390/s17040905.

[91] R. Barth, J. IJsselmuiden, J. Hemming, and E. J. V. Henten, “Data synthesis methods for semantic segmentation in agriculture: A Capsicum annuum dataset,”

Comput. Electron. Agric., vol. 144, pp. 284–296, 2018, doi: 10.1016/j.compag.2017.12.001.

[92] H. Mureşan and M. Oltean, "Fruit recognition from images using deep learning," *Acta Univ. Sapientiae, Inform.*, vol. 10, no. 1, pp. 26–42, 2018, doi: 10.2478/ausi-2018-0002.

[93] A. Torralba, B. C. Russell, and J. Yuen, "LabelMe: Online Image Annotation and Applications," *Proc. IEEE*, vol. 98, no. 8, pp. 1467–1484, 2010, doi: 10.1109/JPROC.2010.2050290.

[94] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, pp. 5967–5976, doi: 10.1109/CVPR.2017.632.

[95] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, "Context Encoders: Feature Learning by Inpainting," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2536–2544, doi: 10.1109/CVPR.2016.278.

[96] T. C. Wang, M. Y. Liu, J. Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8798–8807, doi: 10.1109/CVPR.2018.00917.

[97] L. E. Ortiz, V. E. Cabrera, and G. MG, "Depth Data Error Modeling of the ZED 3D Vision Sensor from Stereolabs," *Electron. Lett. Comput. Vis. Image Anal.*, vol. 17, no. 1, pp. 1–15, 2018, [Online]. Available: https://elcvia.cvc.uab.es/article/view/v17-n1-ortiz.

[98] M. L. Comer and E. J. D. III, "Morphological operations for color image processing," *J. Electron. Imaging*, vol. 8, no. 3, pp. 279–289, 1999, doi: 10.1117/1.482677.

[99] A. S. Kornilov and I. V Safonov, "An Overview of Watershed Algorithm Implementations in Open Source Libraries," *J. Imaging*, vol. 4, no. 10, 2018, doi: 10.3390/jimaging4100123.

[100] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: A Database and Web-Based Tool for Image Annotation," *Int. J. Comput. Vis.*, vol. 77, no. 1, pp. 157–173, 2008, doi: 10.1007/s11263-007-0090-8.

[101] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT++ Better Real-Time Instance Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 1108–1121, 2022, doi: 10.1109/TPAMI.2020.3014297.

[102] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, 2020, doi: 10.1109/TPAMI.2018.2858826.

[103] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, 2017, doi: 10.1109/TPAMI.2016.2572683.

[104] W. Liu *et al.*, “SSD: Single Shot MultiBox Detector,” in *ECCV 2016. Lecture Notes in Computer Science*, 2016, pp. 21–37.

[105] Z. Shang, X. Wang, Y. Jiang, Z. Li, and J. Ning, “Identifying rumen protozoa in microscopic images of ruminant with improved YOLACT instance segmentation,” *Biosyst. Eng.*, vol. 215, pp. 156–169, 2022, doi: https://doi.org/10.1016/j.biosystemseng.2022.01.005.

[106] Z. Zhao *et al.*, “Large scale instance segmentation of outdoor environment based on improved YOLACT,” *Concurr. Comput. Pract. Exp.*, vol. n/a, no. n/a, p. e7370, 2022, doi: https://doi.org/10.1002/cpe.7370.

[107] V. Mohan and P. Morasso, “How Past Experience, Imitation and Practice Can Be Combined to Swiftly Learn to Use Novel ``Tools'': Insights from Skill Learning Experiments with Baby Humanoids,” in *Biomimetic and Biohybrid Systems*, 2012, pp. 180–191.

[108] A. A. Bhat and V. Mohan, “How iCub Learns to Imitate Use of a Tool Quickly by Recycling the Past Knowledge Learnt During Drawing,” in *Biomimetic and Biohybrid Systems*, 2015, pp. 339–347.

[109] M. Zak, “Terminal attractors for addressable memory in neural networks,” *Phys. Lett. A*, vol. 133, no. 1, pp. 18–22, 1988, doi: https://doi.org/10.1016/0375-9601(88)90728-1.

[110] A. A. Bhat, S. C. Akkaladevi, V. Mohan, C. Eitzinger, and P. Morasso, “Towards a learnt neural body schema for dexterous coordination of action in humanoid and industrial robots,” *Auton. Robots*, vol. 41, no. 4, pp. 945–966, 2017, doi: 10.1007/s10514-016-9563-3.

[111] G. W. Humphreys *et al.*, “The interaction of attention and action: From seeing action to acting on perception.,” *Br. J. Psychol.*, vol. 101, pp. 185–206, 2010, doi: 10.1348/000712609X458927.

[112] V. Klema and A. Laub, “The singular value decomposition: Its computation and some applications,” *IEEE Trans. Automat. Contr.*, vol. 25, no. 2, pp. 164–176, 1980, doi: 10.1109/TAC.1980.1102314.

[113] J.-R. Xiao, P.-C. Chung, H.-Y. Wu, Q.-H. Phan, J.-L. A. Yeh, and M. T.-K. Hou, “Detection of Strawberry Diseases Using a Convolutional Neural Network,” *Plants*, vol. 10, no. 1, 2021, doi: 10.3390/plants10010031.

[114] S. Zhao, J. Liu, and S. Wu, “Multiple disease detection method for greenhouse-cultivated strawberry based on multiscale feature fusion Faster R_CNN,” *Comput. Electron. Agric.*, vol. 199, p. 107176, 2022, doi: https://doi.org/10.1016/j.compag.2022.107176.

[115] N. P. Mahalik and A. N. Nambiar, “Trends in food packaging and manufacturing systems and technology,” *Trends Food Sci. Technol.*, vol. 21, no. 3, pp. 117–128, 2010, doi: https://doi.org/10.1016/j.tifs.2009.12.006.