Real-time 3D Detection and Tracking of Broccoli Crops for Autonomous Selective Robotic Harvesting

PhD Thesis

Hector A. Montes

UNIVERSITY OF LINCOLN

School of Computer Science
Lincoln Centre for Autonomous Systems

Thesis supervisor:
Dr Grzegorz Cielniak

Abstract

This PhD research is concerned with robotic perception using state-of-the-art 3D sensors for real-time detection and tracking of broccoli crops. Reliable and robust systems to detect and track vegetables in open farm fields are crucial for autonomous selective harvesting robots. However, automated harvesting of broccoli heads presents several research challenges for current robotic perception technology including complex and highly cluttered environments, difficult lighting conditions, plant growth rates and weather incidences.

Real-time 3D perception of the complex environment where broccoli plants grow, remains a major challenge and is crucial for the adoption and deployment of reliable autonomous harvesting robots. This requires enhanced 3D processing for detection, spatial localisation and tracking that is both robust and precise. Thus, developing automated methods for selective harvesting capable of accurately identifying and separating broccoli crops from the background would help to increase productivity and to better control production costs.

This thesis focuses on the task of autonomous selective harvesting of broccoli crops and tackle the specific problems of broccoli head detection and tracking in real-time using low-cost 3D sensors. 3D vision approaches can bring several benefits when addressing the many challenges involved in harvesting, such as detection, localisation, size estimation, occlusion handling and shape analysis. Due to numerous social, political and economical factors, the interest in automation of agriculture has grown worldwide. Thus, building autonomous systems capable to detect, analyse and harvest crops is rapidly becoming a necessity.

Previous works have identified broccoli plants in the field, as well as measure the size of each plant head to determine whether it is suitable for cutting. A more robust solution to this problem is proposed by integrating this previous work based on low-cost 3D structured light cameras. The goal is to ensure that an automated selective

broccoli harvester could be built using a competitively priced imaging system able to deliver the required levels of accuracy, reliability, and scalability. The thesis focuses on perception methods for the detection and tracking of broccoli crops.

Four clustering methods as part of a broccoli detection pipeline that process RGB-D datasets collected with low-cost sensors were developed for this thesis. These methods achieve high performance in detection by exploiting the organised structure of the 3D data being processed. Similarly, aided by current progress in machine learning, an efficient and effective approach using 3D information for detection and segmentation of broccoli heads based on a Convolutional Neural Network architecture was implemented. The system achieved a high performance in terms of accuracy, segmentation, and localisation, with a better generalisation for the most difficult datasets at high processing speeds.

Finally, a tracking method of broccoli heads based on a Particle Filter was developed to complement the detection pipeline. Crop detection and tracking are together an important part of autonomous selective harvesting as they play a key role in the accuracy of the harvester's cutting system. Tracking is also an important task to uniquely identifying crops with applications in other important agricultural tasks such as crop mapping and produce counting for crop yield prediction. The approach presented in this dissertation combines a broccoli detector and the particle filter to track multiple crops in a sequence of 3D data frames. The tracking algorithm improves broccoli detections and crop estimates, as predictions are made based on dynamics and measurements. Also, the tracking accuracy is verified based on a classification method that associates detections with tracks over each frame. The framework is efficient as it process 3D data at high frame rates.

Real-time 3D perception of the environment is crucial for the adoption and deployment of reliable autonomous harvesting robots in agriculture. The overall goal of this research is to develop and demonstrate machine learning algorithms able of processing 3D data that deliver high detection and tracking performance of broccoli heads at real-time operation speeds. The experimental evaluation presented in this thesis shows that the proposed methods accurately detect and track the 3D locations of broccoli heads relative to the vehicle at high frame rates, while showing comparable performance against state-of-the-art approaches based on 3D segmentation techniques.

The hypothesis evaluated in this research is that a robotic system using low cost 3D sensors capable of detecting and tracking the target crop, improves harvest success and increases production while decreasing labour and other harvesting related costs.

Acknowledgments

I begin with two words that all men have uttered since the dawn of humanity: thank you[1].

There are a few important people who supported and helped with shaping my research work, my thoughts, my curiosity, and even my multiple mishaps along the way to finally make this thesis happen.

First and foremost, I would like to thank my supervisor Dr Grzegorz Cielniak for being a great pillar throughout these short years of PhD work. His patience, constant guidance, permanent willingness to help; his sound and accurate feedback and criticisms, while not forgetting to be a great guy, are only a few of the things that I hope I managed to learn and will be able to practice myself. Thank you so much, Greg!

Similarly, this work would not be possible without the assent and encouragement from my former advisor Prof Tom Duckett. I will always be most grateful for his mentorship and support throughout my first years at Lincoln; not only for his crucial help, but also for his tireless enthusiasm and patience. This thesis could not have been completed without his advise and support (and without a little arm-twisting).

I would also like to express my gratitude to Prof Simon Pearson for his invaluable support and for being the man behind the scenes who made sure everything moved always forward.

Financially, a big thanks goes to the generosity of the Agriculture and Horticulture Development Board (AHDB) for funding my research work and to Dr Jim Dimmock for having made things run smoothly for me.

Finally, and on top of everything and everyone else, and given that my own words

[1] Octavio Paz – Mexican poet, diplomat and Nobel Laureate in Literature. *In Search of the Present.* Nobel Lecture, December 8, 1990.

will never suffice to express my gratitude towards my mom; I take this tiny space to humbly ask her to extend her arms, and generously provide the shelter that her presence has ever been to me –though an ethereal memory now–. She is the only person to whom I will be eternally grateful for simply having been there and for simply having been herself. *Gracias*.

To Duvy Duvy, who has always wanted me to be happy.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation and Context

Harvesting food is an activity performed since the origins of humankind. Agriculture has evolved from traditional gathering-based communities into the modern large-scale farming industry it has become today. All the changes the sector has experienced have helped farms to undergo a rapid development through mechanisation and technological innovations, as a means to meet the growing demand of agriculture products [Valin et al., 2014]. Among the various farm activities, harvesting is an important operation that often employs a considerable labour force and large mechanised resources. It is an operation in which the produce is removed from the plant once is fully grown and developed, and it represents the last cultivation task for the crop [Erkan and Dogan, 2019].

The use of machinery in agriculture has a long history and technological innovation has always been fundamental for its progress. Introduction of automation has lowered operations costs, managed the crops in shorter periods of time, reduced labour involvement, improved the quality of the produce, and better controlled environmental effects. Even though many farm operations can be conducted using commercial mechanised solutions with various degrees of autonomy, autonomous harvesting operations have not yet gained similar levels of development, commercialisation and deployment [Kootstra et al., 2021].

In modern farming practices, a large assortment of machines have been developed to harvest various types of crops and the harvesting method depends entirely on the type of planted crop. Two approaches can be readily compared when harvesting, namely, *mass* or *slaughter harvesting*, *i.e.,* removal of the produce during the harvesting season in one pass, and *selective harvesting*, *i.e.,* cutting only those parts of the crop that meet certain criteria [Bachche, 2015].

Nowadays, crops of planted fruits and vegetables are harvested either manually or by a machine. Some crops produce a single batch of edible product (*e.g.*, fruits destined for processing, tuberous roots, wheat or maize), while others produce several batches during a growing season (*e.g.*, delicate produce such as berries, or vegetables and fruits for fresh consumption). The former are fairly equal in ripeness and become ready for harvesting at the same time, while the latter typically ripen at different rates, thus requiring more than one harvesting pass and also requiring a more careful treatment of the produce to preserve their value and keep them suitable for the standards of today's fresh market. In consequence, these last crops, also referred to as "high-value crops" or "specialty crops", require selective harvesting for they need to be individually assessed to decide which crop is ready to be harvested.

Harvesting of high-value crops is still done manually throughout the world by human labour, as machines are not yet able to both efficiently and accurately perform autonomous selective harvesting at the scale required by the modern farming industry [Zhang and Karkee, 2021]. The considerable labour necessary to carry out this task, increases harvesting expenses and a major reason for a crop to be classified as high-value is, precisely, the intense labour required [Bac et al., 2014]. These high costs are a major motivation driving the development of autonomous selective harvesters for high-value crops. Even though other major motivations involve factors such as social, political and environmental issues, the main goals of growing fresh fruit and vegetables remain keeping the quality high while minimising production costs [Duckett et al., 2018].

1.1.1 Growing and Harvesting Broccoli

Broccoli is a high-value vegetable in the cabbage family that belongs to the *Brassica Oleracea* plant species. Brassica vegetables comprise a large number of closely related plants including some of the world's most commonly cultivated vegetables, such as kale, Brussels sprouts, cabbage, cauliflower and, certainly, broccoli . Even though the plant species originated in the Mediterranean region, cultivated brassicas have now a worldwide distribution, from the tropics to the Arctic Circle [Fahey, 2016].

Broccoli is a plant with abundant dark or bright green edible flower buds arranged in a compact head of a tree-like shape. Broccoli heads grow surrounded by large leaves and are harvested before the flower buds open. Over 26 million tons are produced worldwide prominently by China, India, USA, Spain, Mexico and Italy [FAO, 2022]. The interest in its cultivation has grown over the years due to genetic improvement programmes developed in several countries and to the nutritional compounds contained in the crop that have increased its consumption [Maggioni et al., 2010].

The components of the vegetable depend on many factors such as diversity, harvest period, environment where they grow, processing and even cooking conditions. Broccoli contains low fat, minerals, fibre, carotenoids, folic acid, high levels of vitamin K, and well known antioxidants enzymes such as vitamins C and E [Fahey, 2016].

Broccoli is eaten raw or cooked. A diet rich in broccoli and other Brassica vegetables is linked to a reduced risk of several human cancers and to prevent and treat malignant and degenerative diseases [Baenas and Wagner, 2019, Nagraj et al., 2020]. Also, in Brassica products, there is a high amount of folate that reduces vascular disease and neural tube defect risk [Sanlier and Guler Saban, 2018].

Broccoli varieties have been cultivated for many centuries and have been extensively crossed and hybridised. One consequence of the methods used to breed broccoli is that the heads grow and ripe at a dissimilar pace, as shown in Figure 1.1. This makes them difficult to harvest. Moreover, all cultivated broccoli is selectively harvested by hand, relying on visual grading to estimate whether a head can be cut [Bender et al., 2020]. As a result, only 50% to 60% of broccoli heads are able to be harvested economically. The remainder either mature too quickly and have to be left in the field, or they are slow to grow and do not reach full maturity during the harvesting season. This variation is common in both organic and non-organic broccoli crops [Orzolek et al., 2012].

Figure 1.2 shows two harvesting operations of broccoli heads as is currently practiced in farm fields.

Slaughter or mass harvesting is not a productive option when it comes to harvest broccoli, as it potentially produces large quantities of unmarketable broccoli heads.

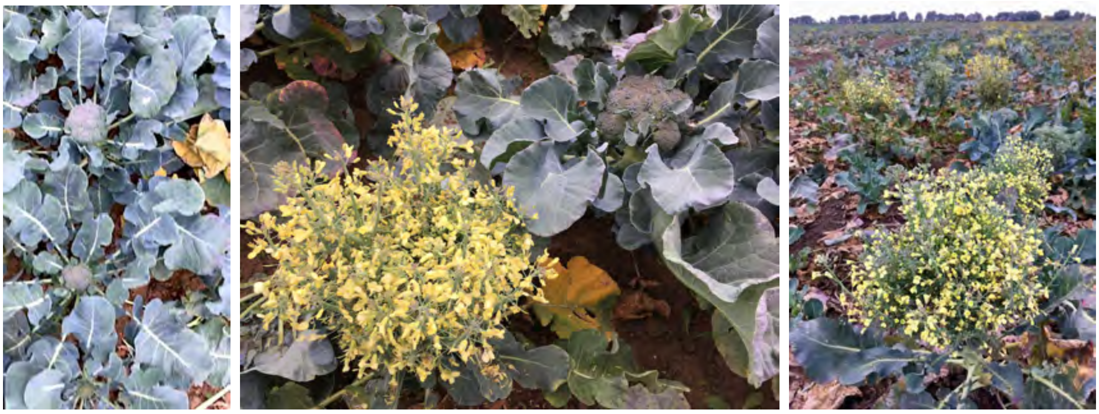


Figure 1.1: Broccoli heads grow and mature at different rates. On the left, seen at the bottom, one head has already been cut, while two others have not yet reached a marketable size. The picture in the middle shows a broccoli head that has already flowered next to another head that can still be harvested. On the right, a large proportion of broccoli heads are left in the field after at least two harvesting passes. Best seen in colour.

Selective harvesting, on the other hand, presents its own challenges, because it relies on a subjective assessment by each person cutting the broccoli to decide which head is ready to be cut. All this, on top of the problem of increasingly scarce and more expensive labour due to issues ranging from political pressures to migration dynamics [Duckett et al., 2018].

1.2 Research Problem

Automatic harvesting requires the development of advanced technologies able to deal with the complex and unstructured nature of the external environment of farm fields and the diverse and highly variable products. This complex environment requires the development of advanced and robust machines to selectively harvest a high-value crop such as broccoli.

Harvesting is the final operation of broccoli cultivation and determines the vegetable quality. It is important to harvest broccoli heads at the proper maturity stage to maintain their nutrient and freshness quality. Harvesting broccoli should be done carefully without damaging the produce. Tubers and other crops have been mechanically harvested for many years, mainly because they can endure rougher handling and still maintain quality. Crops like broccoli, however, present different challenges

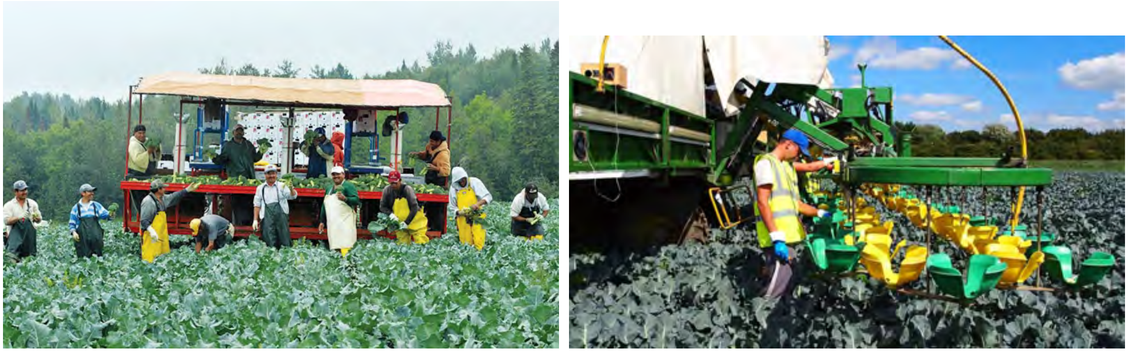


Figure 1.2: Teams of broccoli pickers. The team on the left, cuts the broccoli heads and puts them on the vehicle for sorting and packaging. The team on the right, uses a conveyor system harvest aid to carry the broccoli heads into the vehicle. These are two of the most widely methods used for harvesting brassicas. Best seen in colour.

for mechanised harvesting because of the delicate nature of the product. As a result, most harvesting and packing is still being done by hand with the consequent intensive labour use and time consuming process.

Labour shortages and other production costs have made autonomous robotic harvesters an attractive option to detect, position, and grade broccoli crops to automatically harvest the heads [Kootstra et al., 2020]. Autonomous selective harvesting robots aim to minimise the labour work as well as increasing the speed and accuracy of the harvesting operations. Eventually, they will be able to perform the tasks of the now increasingly scarce human hands [Kootstra et al., 2021]. It is therefore desirable to research and find methods to harvest more frequently, more quickly, more accurately and with less waste to reduce labour and other operation costs [Bac et al., 2014].

Even when full automation of agricultural tasks have been actively developed for some decades now, important tasks such as harvesting or crop yield estimates still rely on intense human labour. In this sense, crop detection and tracking are together an important part of autonomous selective harvesting. They can also increase accuracy of other tasks in agriculture such as produce count, mapping or estimates of crop yields. The result of these tasks can then be used by farmers on the entire agricultural process to make informed decisions on several tasks including planting, harvesting and labour management [Zhang and Karkee, 2021].

Reliable and robust systems to detect and track vegetables in real open farm fields are crucial for harvesting robots. However, selective harvesting of broccoli heads presents several research challenges for current robotic perception technology [Silwal et al., 2021]. Broccoli harvesting robots are required to navigate planted fields and to make detection and cutting decisions of broccoli heads in high dimensional, unstructured, and complex environments where 2D sensors are insufficient. 3D sensing technology would allow an autonomous selective robotic harvester to achieve levels of autonomy that is more limited or not even possible with 2D vision alone. Moreover, current 3D sensor are affordable, reliable, and able to deliver high resolution RGB-D data.

Outdoor field conditions in agriculture present challenges for 2D sensors such as variable lighting conditions and limited view points that can be more reliably handled by 3D sensors. The advantages of 3D include encoding the broccoli head geometry and providing different view points under the cluttered conditions found in farm fields. For instance, using only 2D machine vision techniques may be insufficient for a reliable processing of the multiple occlusions occurring at various depths in the crops of broccoli plants. Also, there are currently a number of problems with 3D reconstruction from 2D images in real-time. In contrast, computational requirements and algorithmic challenges may increase when handling 3D information, along with the large amounts of data required to train robust detectors. Nevertheless, the development of software tools, such as ROS[1] and PCL [Cousins and Rusu, 2011], have contributed to a more efficient and effective processing of 3D information.

Real-time 3D perception of the complex environment where broccoli plants grow, remains a major challenge and is crucial for the adoption and deployment of reliable autonomous harvesting robots. This requires enhanced 3D processing for detection, spatial localisation and tracking that is both robust and precise. Thus, developing automated methods for selective harvesting capable of accurately identifying and separating broccoli crops from the background would help to increase productivity and to better control production costs.

This thesis focuses on the task of autonomous selective harvesting of broccoli crops and tackle the specific problems of broccoli head detection and tracking in real-time using low-cost 3D sensors.

[1] http://wiki.ros.org/agriculture

1.3 Proposed Solution Frameworks

This PhD research is concerned with robotic perception using low-cost 3D sensors for real-time detection and tracking of broccoli crops. The goal is to ensure that an automated selective broccoli harvester could be built using a competitively priced imaging system able to deliver the required levels of accuracy, reliability, and scalability.

Four detection methods that operate at high frame rates on datasets collected with low-cost RGB-D sensors have been developed for this thesis. These methods achieve high performance in detection by exploiting the organised structure of the 3D data being processed as well as a processing time of 68.7 ms for the fastest and 106.5 ms per frame for the slowest. Similarly, an approach using 3D information for detecting broccoli heads based on Convolutional Neural Networks (CNN) was developed. This method achieves an even higher inference time at processing speeds of 50∼60 frames per second, making it even more suitable for autonomous robotic harvesting and other farming operations.

The dissertation also presents a tracking method of broccoli heads based on a Particle Filter. Crop detection and tracking are together an important part of autonomous selective harvesting as they play a key role in the accuracy of the harvester's cutting system. The approach combines a broccoli detector and the particle filter to track multiple crops in a sequence of 3D data frames. The tracking accuracy is verified based on a classification method that associates detections with tracks over each frame.

Figure 1.3 depicts a general overview of the detection and tracking framework presented in this dissertation. Both detection and tracking in autonomous selective harvesting systems determine the location of each crop and select appropriate cutting points using 3D sensors. The 3D localisation allows a cutting point and a grasp to be more readily determined by the harvester. Sensing and 3D data processing techniques are one of the fundamental components of an autonomous robotic harvester as it plays a key role in its design criteria and affects the cost, type of actuation system, and speed [Karkee et al., 2021].

The hypothesis evaluated in this research is that a robotic system using low cost 3D sensors capable of detecting and tracking the target crop, improves harvest success and increases production while decreasing labour and other harvesting related costs.

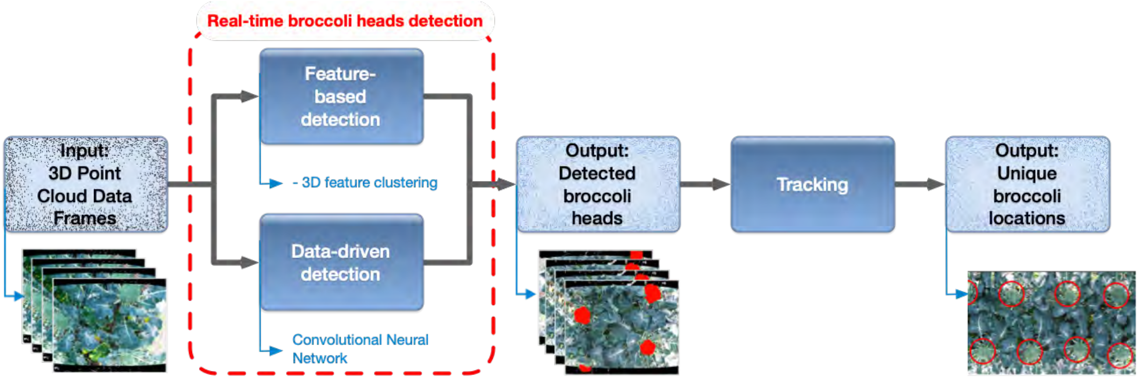

Real-time broccoli heads detection
Input: 3D Point Cloud Data Frames
Feature-based detection
- 3D feature clustering
Data-driven detection
Convolutional Neural Network
Output: Detected broccoli heads
Tracking
Output: Unique broccoli locations

Figure 1.3: General overview of the detection and tracking framework presented in this thesis. The input data are the point cloud frames acquired by the RGB-D sensor. The output is the locations of the broccoli heads detected on each input frame.

In line with this hypothesis, the overall goal of this dissertation is to develop machine learning methods for processing 3D data that deliver high precision detection and tracking of broccoli heads and perform in real-time.

From the hypotheses formulated above, follows the main research question: Which novel and proven 3D point cloud processing methods can efficiently improve detection and tracking performance for automated selective harvesting by coping with broccoli variation in real farm field conditions?

To address both this question and the goal of this research, the major work developed could be divided into three main objectives: (1) To develop 3D clustering algorithms applied to organised point clouds to extract complex 3D features of broccoli plants as part of a detection pipeline designed for autonomous robotic selective harvesting. (2) To implement a technique for broccoli heads detection and segmentation based on a deep learning algorithm also applied to organised 3D point clouds that performs at high speed frame rates. (3) To develop a tracking method by including 3D real-time detection methods into a 3D model-based particle filter tracking approach.

The results show that the proposed methods are capable of accurately detecting and tracking the 3D broccoli crop locations relative to the vehicle at high frame rates, while achieving comparable performance to state-of-the-art approaches based on 3D segmentation techniques.

1.4 Contributions

This PhD dissertation presents methods to detect and track multiple crops of broccoli plants in sequences of 3D data at high frame rates. All these approaches extensively exploit the organised nature of the point clouds originated from the RGB-D sensors and, thus, are suitable for being incorporated into an autonomous selective harvester. The main contributions of the thesis are:

1. Feature-based broccoli detection model: We explore four different methods for detecting broccoli heads from point cloud data. To effectively detect the broccoli crops we develop:
   - an efficient 3D feature-based detection system able to operate in real-time and process depth information at high frame rates. The system is able to generalise across variations in crop appearance while keeping high detection success rates at average processing times as low as 68.7 ms per frame.
   - four efficient and effective algorithms for clustering point clouds that extracts regions based on euclidean distances, normal vector angular properties, surface curvatures and depth boundaries. The rich information contained in the 3D data is used for solving the clustering and segmentation problem of broccoli crops.
2. Data-driven broccoli detection model (deep learning model). To further improve the detection of broccoli crops we develop:
   - a technique that uses a Convolutional Neural Network (CNN) architecture applied to organised 3D point clouds for broccoli detection and segmentation.
   - improvement of the generalisation capabilities of the CNN-based detector through data augmentation techniques. The system achieved a high performance in terms of accuracy, segmentation, and localisation, with a better generalisation for the most difficult datasets at processing speeds of 50∼60 frames per second.
3. Tracking: We adopt a tracking algorithm by including the 3D feature-based real-time detection methods into a 3D model-based particle filter tracking

approach. The method introduces:

- an effective tracking algorithm for improving broccoli detections and crop estimates, as crop predictions are made based on dynamics and depth information measurements.
- an efficient particle filter framework that restricts the search of crops in subsequent frames and reduces the number of false negatives a detector can produce on its own.
- an efficient system that runs in real-time and processes 3D data at high frame rates. The algorithm adds 28 ms when it processes every frame together with the detector, and adds 9 ms when the detector is used every two frames.

4. Evaluation: We present an extensive experimental evaluation of the detection and tracking methods on datasets of different broccoli varieties collected from two countries under real field conditions.

1.5 Publications

Part of the material and results included in this thesis has been presented in several forums. The following is a list of complete publications written during the course of this PhD work.

- Montes, H. A., Cielniak, G., and Duckett, T. (2019). Model- based 3D point cloud segmentation for automated selective broccoli harvesting. In *The 20th Towards Autonomous Robotic System (TAROS) Conference*, pages 448–459.
- Montes, H. A., Le Louedec, J., Cielniak, G., and Duckett, T. (2020). Real-time detection of broccoli crops in 3D point clouds for autonomous robotic harvesting. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10483–10488.
- Louedec, J. L., Montes, H. A., Duckett, T., and Cielniak, G. (2020). Segmentation and detection from organised 3D point clouds A case study in broccoli head detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 285–293.

- Montes, H. A. and Cielniak, G. (2022). Multiple broccoli head detection and tracking in 3D point clouds for autonomous harvesting. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence, workshop on AI for Agriculture and Food Systems.*

1.6 Organisation of the Thesis

The thesis starts by presenting the current state of the art in the related research area: detection and tracking of multiple crops of broccoli plants in Chapter 2.

Following the exploration of existing work, Chapter 3 presents four clustering methods based on 3D features that operate at high frame rates on datasets collected with low-cost RGB-D sensors in open farm field conditions. While sharing the goal with previous works, this thesis approaches the problem of broccoli detection by extensively investigating methods to explore the spatial 3D features of the data points observed in the crop of interest. These methods achieve high performance in detection by exploiting the organised structure of the 3D data being processed. A brief note on the 3D point clouds datasets and their ground truth annotations is also included in this chapter.

Chapter 4 details an approach using 3D information for detecting broccoli heads based on Convolutional Neural Networks (CNN) architecture to organised point clouds for detection and segmentation. The method improves its generalisation capabilities through data augmentation techniques and achieves a high performance in terms of accuracy, segmentation, and localisation, with a better generalisation on the most difficult datasets at processing speeds of 50 frames per second.

The last technical chapter of this dissertation, Chapter 5, presents and evaluates a tracking method of broccoli heads based on a Particle Filter. The approach combines a broccoli detector and the particle filter to track multiple crops in a sequence of 3D data frames. The framework is efficient as it restricts the search of crops in posterior frames and process 3D data at high frame rates.

Finally, we present our conclusions, possible improvements, and suggest directions for future work in Chapter 6.

2 Related Work

Autonomous selective harvesting presents numerous challenges, such as identification of crops, localisation, segmentation and analysis, which requires fast operating speeds. Automated harvesting systems usually consist of three independent systems [Bachche, 2015, Kootstra et al., 2020, Vrochidou et al., 2022]:

1. a recognition system to detect and locate the product,
2. a picking system to perform grasping and cutting operations,
3. a navigation system to allow the robot to move around the cultivated crop plants.

One major challenge in autonomous harvesting is to reliably identify and locate the crop from the rest of the plant and other elements in the background at real-time operating speeds. One of the first and common approaches has been to detect crops using 2D images. This can be promptly perceived in the wealth of techniques based on computer vision available in the literature [Jimenez et al., 2000, Bachche, 2015, Hamuda et al., 2016, Zhao et al., 2016, Kootstra et al., 2021]. For the particular case of broccoli crops, some approaches have used RGB images to separate the broccoli head from the soil and other plant parts, whilst few others have used native 3D sensing technology.

This chapter addresses the most relevant work for detecting crops of broccoli plants, the latest achievements of some researchers, as well as some other relevant methods developed for similar high-value crops. In addition, the chapter presents the justification of this research and the need of effective and efficient harvesting systems for crops of broccoli plants.

2.1 Early Efforts on Fruits and Vegetables Detection

Since the first publication of computer vision approaches for some agricultural tasks nearly four decades ago, methods that process digital images have seen it as a fundamental sensing technology [Liakos et al., 2018, Silwal et al., 2021]. In these detection systems, different types of cameras, such as CCD, infrared, or multispectral cameras are used to capture the images often combined with artificial lighting systems [Zhao et al., 2016]. The images were then processed by algorithms that computed a wide variety of feature attributes (*e.g.*, shape, size, edges or colour) to detect and locate fruits and vegetables. These algorithms discriminated the produce of interest from other parts in the background, output their corresponding locations and, in some cases, their orientations or other parts of harvesting interest (*e.g.*, length and angle of the stem) [Mo et al., 2021].

The first published research reports for high-value crops were on a quality assessment system for grading tomatoes [Sarkar and Wolfe, 1985], on sweet-pepper orientation detection [Wolfe and Swaminathan, 1987] and on peach and apple detection in orchards [Sites and Delwiche, 1988]. Most of these methods were based on basic colour or threshold segmentation techniques with the notable drawbacks they entail, as the settings used for some scenarios fail when the environment conditions changed (*e.g.*, lighting conditions, variable backgrounds and colour changes through crop growth or ripeness) [Hamuda et al., 2016].

2.2 Detecting Broccoli Crops

Early research works using broccoli data were developed during the 1990s based on computer vision methods for determining the different maturity stages and sizes of broccoli heads. [Wilhoit et al., 1990] evaluate the feasibility of using a digital image processing system for selecting mature broccoli heads based on size. In this work, a set of 48 images of broccoli plants with a wide range of head sizes were used to test a model based on the grey-level run length method of texture analysis to differentiate between the broccoli heads and their background. Authors found that the model exhibited an exponential relationship between the texture measurement and the broccoli head area. The results indicated an error of less than 1.0 cm in head diameter estimation for a 10 cm diameter head. This size was then used to

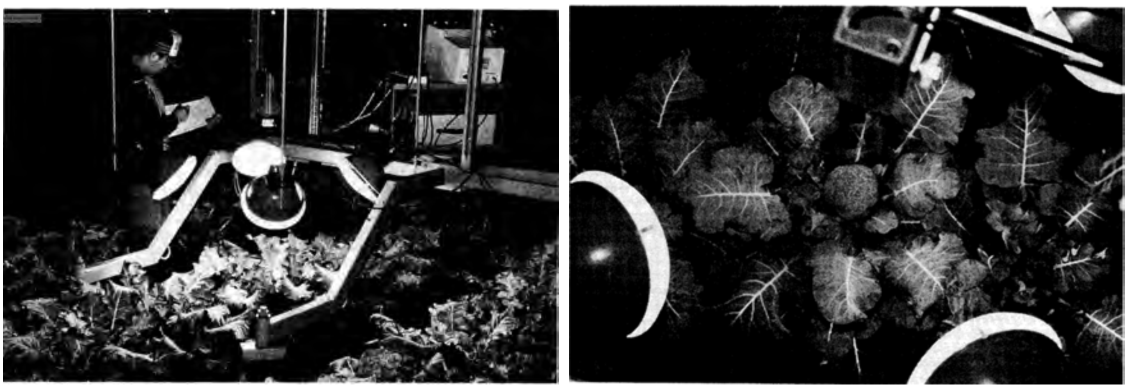


Figure 2.1: The four wheel cart with lighting and camera arrangement constructed to collect the image data used in the experiments reported in [Qiu and Shearer, 1992] and [Shearer et al., 1994]. Images reproduced from [Qiu and Shearer, 1992].

classify the produce into immature and harvestable broccoli heads. The images, however, were collected indoors under controlled lighting conditions and all at the same distance from the camera.

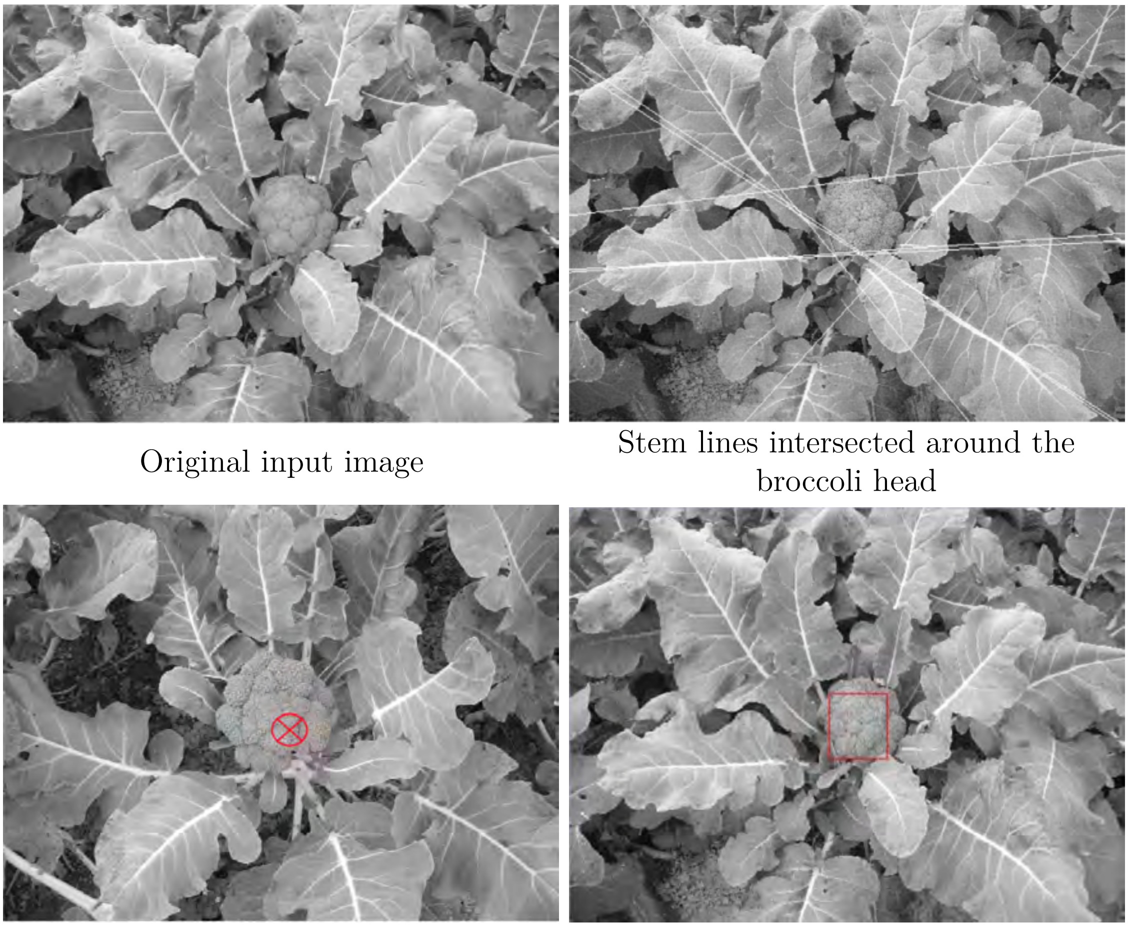
Nonetheless, there is a notable disagreement between experimental controlled conditions and the unstructured and variable environment found in real farm fields. Such real and often harsh conditions produced unacceptable results in a previous work published by [Soule and Sides, 1988]. The experiments reported there measured, under a controlled lighting scheme, the radiant intensity of reflected light to assess the colour reflectance property of a broccoli head to determine its maturity. They concluded, however, that this colour property is not sufficient to accurately determine maturity of the crop.

A method to assess the maturity of broccoli heads based on the analysis of line scan images and the Discrete Fourier Transform (DFT) was developed by [Qiu and Shearer, 1992]. The experimental setup for this work consisted of 160 images from each of three broccoli varieties and only one broccoli head was visible in each frame. The image data was collected at night to better control ambient light using cameras, lights, reflectors, and diffusers assembled on a four-wheel cart towed by a tractor, as shown in Figure 2.1. The DFT was applied to a grey-scale line scan taken of each broccoli head. Their results showed that the method was able to discriminate between various states of maturity by using the average response values of specific frequency bands. The maximum accuracies achieved were 88.1% and 85.0% for individual and multiple varieties, respectively.

Later [Shearer et al., 1994] used the same dataset to determine the maturity of broccoli using a co-occurrence texture analysis method. One distinctive feature of a broccoli head is its texture, caused by a large collection of tiny buds. For this study, a co-occurrence matrix was created from the grey-scale line scan taken of each broccoli head. Texture features were then applied to determine its maturity. The success rate achieved was of 83.1% for discriminating between mature and immature broccoli heads.

A similar texture analysis method was taken up again by [Ramirez, 2006]. This work, however, only considered a small set of 13 RGB images of entire broccoli plants taken in the field, but this time under a variety of natural lighting conditions, camera distances and angles. Still, only one broccoli head was visible in each image, while the main goal remained to distinguish between mature and immature broccoli heads. The method first combined a threshold filter, a Canny edge detector, and a Hough Transform to extract geometric features to approximate lines that can be fitted in the image to find the leaf stems of the plant. To this end, a key component was the Hough Transform. In the images, the stems represent most of the long line segments. A Hough transform line detector operates by finding all white pixels in a binary image and assumes they could be part of a line that lie along the stems of each leaf. The locations of all the lines intersections were averaged and the resulting coordinates were considered the location of the broccoli head. Each detected head was then sized based on analysis of local contrast, *i.e.*, regions of similar contrast are considered to be part of the same measurable area. The maturity of the broccoli head was finally determined based on a co-occurrence matrix texture analysis adopted from [Shearer et al., 1994]. A sequence of sample images of this process is shown in Figure 2.2. Unfortunately, the limited size of the dataset used was too small to draw a conclusion on the applicability of the method in open field conditions.

Tu *et al.* [Tu et al., 2007] published results of a method to grade broccoli heads. The goal was to assess the quality decay of the harvested crops based on the extraction and analysis of feature parameters such as colour and shape. The images were captured under controlled conditions in a chamber box using samples of broccoli heads obtained from a farmer. The system determined the area and roundness



Original input image

Stem lines intersected around the broccoli head

Results of average line intersection

Broccoli head as found by contrast analysis

Figure 2.2: Image sequence reproduced from the experiments reported in [Ramirez, 2006].

as the shape parameters (as shown in Figure 2.3) and extracted the colour features using standard vision techniques. The colour of each head was first captured in RGB and then determined by reflectance mode and expressed in L*a*b* parameters. The colour of the broccoli head surface can be indicated by measuring the resultant colour yellowness percentage using the values of a reference table made by human graders. The quality decay of the broccoli head was then determined by a neural network classifier using these colour and shape features. The highest predicting accuracy achieved by this method was of 93.4%.

Also using RGB images, [Blok et al., 2016] presented a method for detecting and

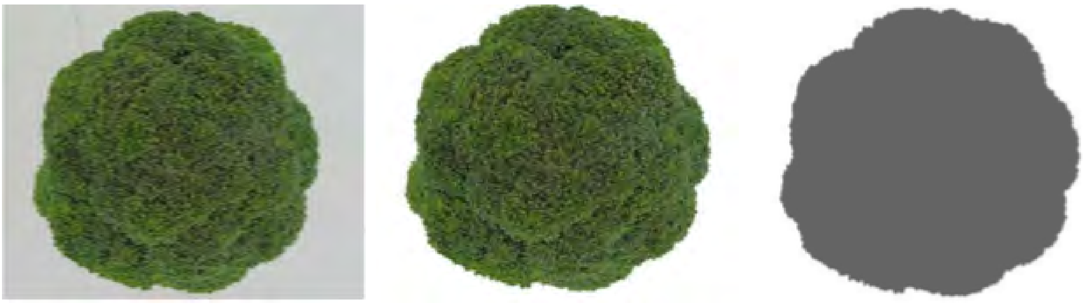


Figure 2.3: Image processing of a broccoli head sample to determine its shape. The first image is the input image. In the second frame the background was cleaned and then thresholded to get the final image from which the shape was estimated. Images reproduced from [Tu et al., 2007], best seen in colour.

sizing broccoli heads based on standard computer vision techniques. 200 images containing 208 broccoli heads were randomly selected from a larger set of 7,008 images of two broccoli varieties collected using an RGB camera and artificial light inside a purpose-built enclosure to block natural light. Only one –or none– broccoli heads were visible on each image at a distance of 50 cm to the camera. The method segmented an image based on texture and colour of the broccoli head appearance. Firstly, the contrast of the image was enhanced to emphasise high frequency areas, followed by a series of filters and morphological operations to fine-tune the image. Then, a set of green-coloured connected components was generated. Lastly, a shape-based feature selection on the connected areas was conducted to separate small non-connecting components from the foreground. In addition, the segmented heads were also sized using circle templates. A sequence of frames depicting this process is shown in Figure 2.4.

The mean image processing time was around 300 ms on an Intel i7 processor at 2 GHz clock speed. The system was part of a prototype harvesting device attached to a modified tractor and was tested in cultivated broccoli fields reaching a precision score of 99.5%, a recall score of 91.2%, and a negative predictive value, *i.e.*, the proportion of negative results, of 69.7%. Authors argued that the sensitivity score was less important than the precision score, as a low precision and a high number of false positives would mean unwanted cutting actions that could potentially damage both the crop and the cutting tool.

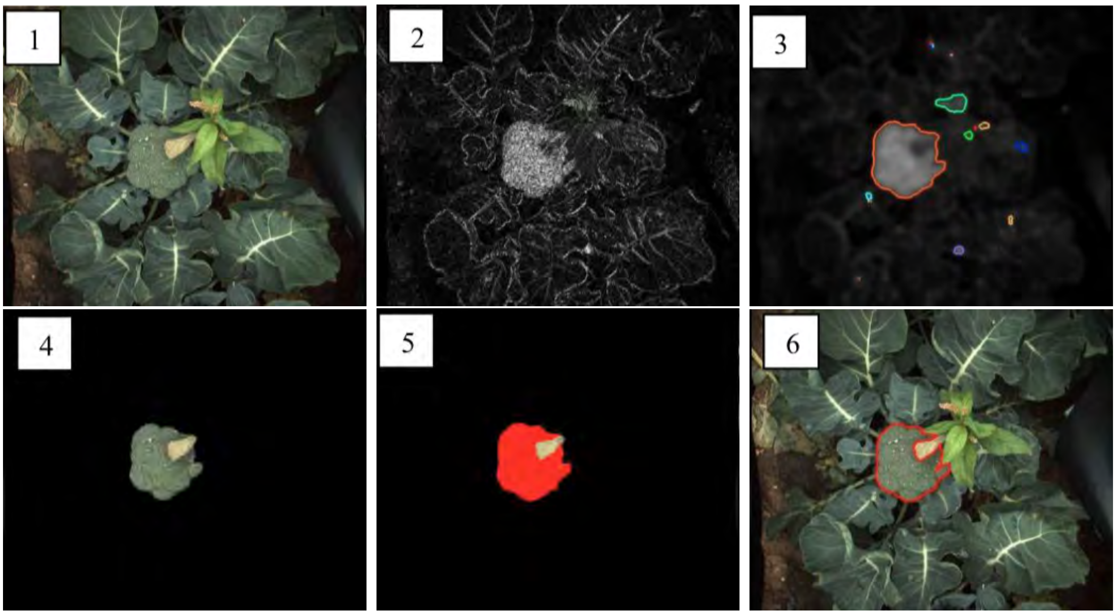


Figure 2.4: The different stages of the texture based image segmentation implemented by [Blok et al., 2016], from the original image (1) to the final segmented broccoli head encircled in red in frame (6). Images reproduced from [Blok et al., 2016], best seen in colour.

In a work extended from [Kusumam et al., 2016], [Kusumam et al., 2017] proposed a system for detecting and locating mature broccoli heads in real farm conditions based on depth images acquired by a low-cost RGB-D sensor. This is the first work, to the best of our knowledge, that investigates the feasibility of using low-cost 3D cameras to identify mature broccoli heads in real planted fields. Other works, however, have also used 3D data for various other crops, *e.g.*, [Barnea et al., 2016] and [Sa et al., 2017] for detecting sweet peppers, [van Henten et al., 2009] for cucumber harvesting, [Gai et al., 2015] for weed discrimination, and [Nguyen et al., 2016] for detecting red and bi-coloured apples. The method presented by Kusuman *et al.* evaluated different 3D features to detect broccoli heads and provided their 3D locations relative to the vehicle. According to their paper, two sets of two broccoli varieties containing 1,769 point cloud frames were selected from a much larger collection of 3D frames captured from fields in the United Kingdom and Spain. Multiple broccoli heads can be seen on each frame in both datasets.

Authors designed a pipeline to process the raw point cloud data that included an outlier removal step, an Euclidean clustering method, a 3D feature descriptor and Support Vector Machines (SVM) classifier to locate the produce. A temporal filter

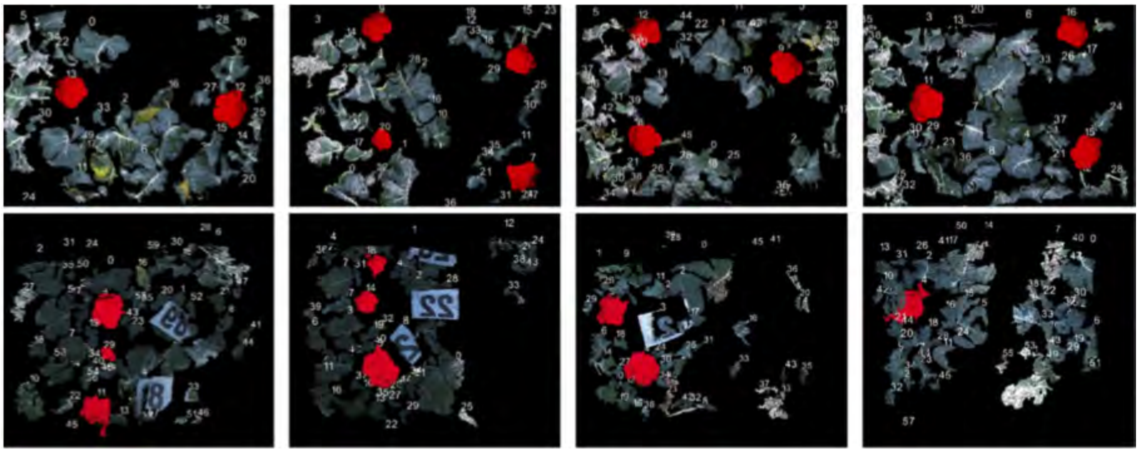


Figure 2.5: The output of the detection pipeline from [Kusumam et al., 2017] on selected frames from the U.K. dataset (top row), and the Spanish dataset (bottom row). The red segments are the positive broccoli heads detections. Images reproduced from [Kusumam et al., 2017], best seen in colour.

was also added to remove false positives and track the detected broccoli heads. After removing point outliers, the Euclidean clustering method also removed the ground plane and group the remaining point cloud segments into clusters. Then a global 3D feature descriptor was extracted from each cluster to train the classifier to distinguish between broccoli heads and other background elements such as leaves or soil. See Figure 2.5 for a sequence of the detection results. Their results showed an average precision score of 95.2% and 84.5% on the datasets collected from fields in the United Kingdom and Spain, respectively. However, on these results is also evident an uneven generalisation performance on the different broccoli varieties considered. Added to this drawback, the average processing time for the method was between 5 and 6 seconds per frame.

The broccoli head sizes were also estimated to determine when a head is ready for harvest using two methods based on a bounding box and a convex hull. The first method estimates the diameter of the head by measuring the limits of the point cloud and only the x axis was considered to measure the width of the head. The second method computes the 3D convex hull of the segment and its centroid. Then the distances between all points and the calculated centroid are measured and the mean value of these distances is the radius of the broccoli head. It was concluded that the distribution of errors for both methods has a similar shape, meaning that

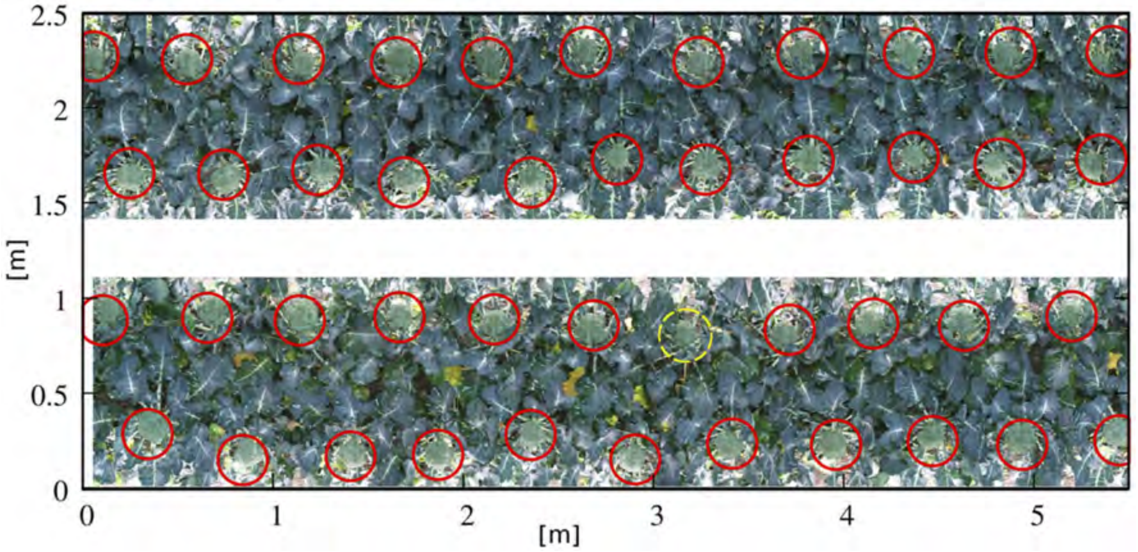


Figure 2.6: A partial map of broccoli locations for the U.K dataset. One misdetection is shown in a yellow circle. Images reproduced from [Kusumam et al., 2017], best seen in colour.

both size estimation methods can be applied at various sensor angles without a significant change in the results. Finally, the temporal filtering step performs an association between the current and the previous frames using the centroids and sizes of the heads detected. The algorithm then finds the closest broccoli, in terms of Euclidean distance and size measured. With this simple scheme, broccoli heads are tracked and 3D maps of the planted broccoli rows can be built, as seen in Figure 2.6.

Similarly to the published work reviewed up to this point, a number of deep learning techniques have been used to process and better understand the large datasets produced in automated agricultural research. A summary of relevant works is presented in the following section.

2.2.1 Deep Learning for Broccoli Detection

Over the past several years, deep learning techniques have been successfully applied in various fields and have also gained momentum in agriculture [Kamilaris and Prenafeta-Boldú, 2018]. These techniques have achieved both high classification performance and real-time execution. However, deep learning approaches are data hungry, as the complexity of problems require large amounts of training data and a considerable annotation effort, both often not readily available and usually expensive and time consuming to acquire. More often than not, this is also true for any other Machine Learning technique, as finding enough labeled data has always been a major challenge in the field.

[Bender et al., 2020] reported experiments on semantic segmentation and object detection of broccoli and cauliflower plants. The real goal of this work, though, was to present a dataset containing weekly scans of cauliflower and broccoli plants in a ten week growth cycle from transplant to harvest. This dataset was collected in Australian farm fields along with ground truth measurements taken by hand directly from the field. The datasets were collected using their own robot called *Ladybird*, a multipurpose robotics platform specifically designed for agricultural operations.

The robot was equipped with an imaging canopy to block direct sunlight. Inside the canopy, two RGB cameras were installed arranged in a stereo configuration along with a thermal camera positioned between the stereo pair. Also, the captured scene was illuminated using a high-powered strobe. A hyper-spectral line-scanning camera was also installed outside the imaging canopy together with a flight-calibration panel to correct for external light conditions. As a result, the dataset contained multispectral high-resolution stereo images.

The hyper-spectral data was compiled into *hypercubes* and used for semantic segmentation using a *Gaussian Process* classifier under the assumption that each spatial unit within the hypercube could be classified independently into broccoli, weed, or soil. An individual model was built for four weeks and 1,000 instances were labeled for each class in each week. Half the data were randomly sampled for training and the remaining data were used for testing. The performance of the model averaged a F-score of 94.5% of all possible train-test combinations of the instances used from each week.

Additionally, authors demonstrated an object detector of cauliflower and broccoli plant instances using a Faster Regional Convolutional Neural Network (R-CNN) with a Resnet-101[1] feature extractor pre-trained on the COCO dataset [Lin et al., 2014]. Data for this task was generated by labelling 1,248 images from the left RGB

[1] http://www.image-net.org

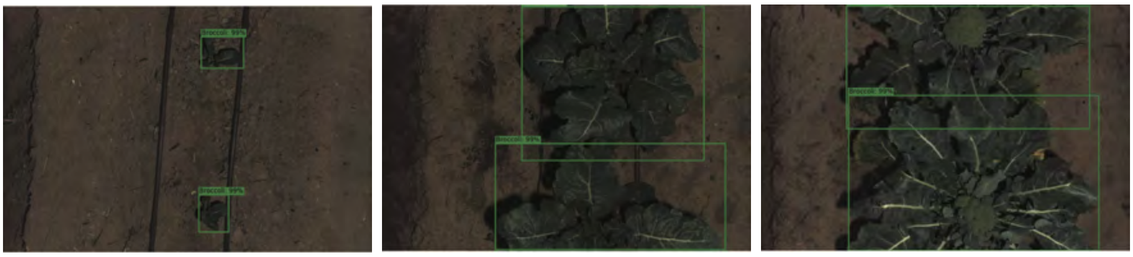


Figure 2.7: Examples of positive detections of the Faster R-CNN broccoli plant detector at different weekly stages of growth. Images reproduced from [Bender et al., 2020], best seen in colour.

camera. The images were corrected for vignetting and flash patterns and downsampled to 1080×720 pixels. Then the annotated images were randomly sampled and partitioned into 60% for training and 40% for testing.

The performance of the Faster R-CNN object detector was evaluated using the PASCAL visual object classes challenge metrics [Everingham et al., 2010]. The average precision is calculated in these metrics for each class and true positives are recorded as detections with an intersection over union (IOU) greater than 50 and 75%. The object detector applied to the colour imagery produced a average precision score of 94.30% for broccoli plants at 50% IOU and 89.51% at 75% IOU. Example detections of broccoli crops at different stages of growth are shown in Figure 2.7.

These experiments were presented to demonstrate how the collected dataset can be used. However, these results were for the entire broccoli (and cauliflower) plant and not for individual heads, which is necessary for autonomous selective harvesting operations.

Also using a deep learning framework, [Zhu et al., 2018] presented a method for vegetable images classification based on the AlexNet CNN model [Krizhevsky et al., 2017]. The goal of this research was the automatic picking and sorting of fresh vegetables. To this end, authors used a dataset from ImageNet[2] to classify five categories of vegetables: broccoli, pumpkin, cauliflower, mushrooms and cucumber. The dataset was enlarged to improve training by creating rotated versions of the original images, which resulted in 24,000 images of the five categories of vegetables.

[2] http:// www.image-net.org

All images were resized to 80×80 pixels and only one head was visible of the two brassica crops considered, *i.e.*, broccoli and cauliflower. Colour and shape features of the images were extracted for training the CNN and then compared to a BP neural network and a SVM classifiers. The expanded dataset was split into 80% for training and 20% for testing and the performance of the CNN achieved an accuracy rate of 92.1%. However, performance results on the individual categories of vegetables were not published. Meanwhile, [Zhou et al., 2020] presented an improved CNN ResNet model for segmenting broccoli heads from RGB images. In this work, a dataset of 506 images were acquired using a Canon EOS 90D camera under controlled conditions. The dataset was later increased to 6,000 images through common data augmentation techniques. A yield estimation model was built based on the number of segmented pixels and a pixel weight value achieving a precision of 89.6% and a recall of 87.9%. In addition, a Particle Swarm Optimisation algorithm and the Otsu method were used to grade the quality of broccoli heads according to a standard proposed by the authors. Even though the results reported by [Zhu et al., 2018] and [Zhou et al., 2020] compare poorly with other published works using 2D images, they are included in this review for the sake of completeness.

Seeking to develop a robot that can selectively harvest broccoli heads, [Blok et al., 2020] presented a detection method using the Mask Region-based CNN model [He et al., 2017]. In their experiments, images from three different broccoli varieties were collected in two countries using a prototype tractor robot not yet fitted with a cutting tool. The robot was equipped with an imaging system that collected top view images of one row of broccoli plants. Because the robot collected images in fields in the Netherlands and in the United States, two different image acquisition systems were used, each equipped with a different RGB camera and illumination setup inside a shrouded box to achieve uniform illumination. Also, a stereo vision camera was installed to acquire depth images to estimate the size of the broccoli heads. 26,000 images of two broccoli varieties were collected in the Netherlands during four consecutive years. An additional set of 14,000 images of another variety were later acquired on one broccoli field in the United States. From the dataset, 1,000 images with broccoli heads were randomly selected for each of the three varieties. The 3,000 images were then resized to 1024×1024 pixels and all visible broccoli heads of various sizes were manually annotated. Each variety subset was randomly divided into one training set of 600 images, one validation set of 100 images and three test sets of 100 images each.

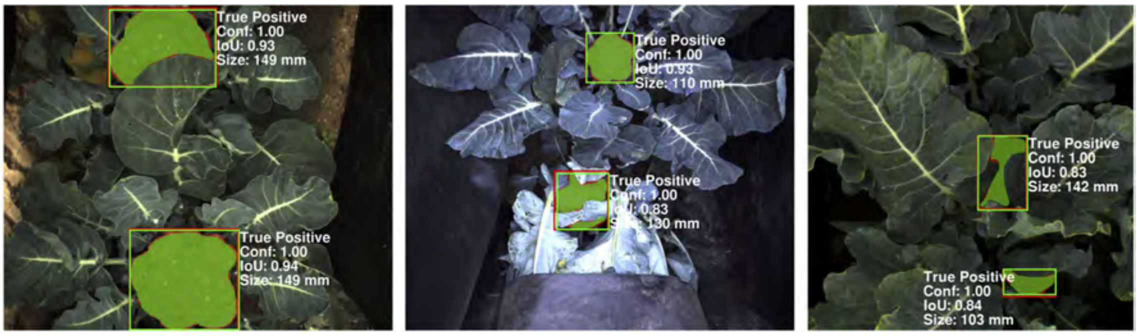


Figure 2.8: Examples of positive detections of the Mask R-CNN broccoli head detector. Images reproduced from [Blok et al., 2020], best seen in colour.

One added goal on this work was to detect broccoli heads from any variety, so authors hypothesised that the developed method can improve its generalisation performance through network simplification and data augmentation. Three geometric transformations, *i.e.*, rotation, cropping, and scaling, were used to produce a better image generalisation than just light, colour, and texture transformations.

The recall and precision detection performance of the Mask R-CNN on their own dataset was of 98.7% and 99.1%, respectively, on images showing only broccoli heads of harvestable size (between 10 and 15 cm). However, recall and precision dropped to 90.0% and 98.2% on images that also included broccoli heads of small (<10 cm) and big (>15 cm) sizes. Detection results were also reported on the broccoli dataset from [Bender et al., 2020], on which the Mask R-CNN system was not previously trained. On this dataset, the method achieved an equal recall and precision rate of 99.4% on images of harvestable broccoli heads, and had an slight recall decrement of 98.8% and a similar precision score of 99.3% on broccoli heads of all sizes. Overall, on both datasets the method achieved recall and precision scores of 99.0% and 99.3% of harvestable broccoli heads, which also decreased to 93.9% and 98.7%, respectively, on images with heads of all sizes. This performance decrement was mainly due to the recall rate on the small broccoli heads, which indicates that system can still be improved, especially for applications when sizing the broccoli heads is needed, *e.g.*, crop yield estimation, plant phenotyping, crop mapping or harvesting. The maximum processing time of the system was 0.27 seconds per image. Example detections of broccoli heads of the Netherlands dataset are shown in Figure 2.8.

Later, [Blok et al., 2021] compared this same Mask R-CNN model with an Occlusion Region-based CNN (ORCNN) architecture [Follmann et al., 2019]. The goal was to perform detection and instance segmentation of broccoli heads, as well as estimating their sizes even when the heads were heavily occluded. The images were collected on fields from two countries where different broccoli varieties were grown. In addition to the same USA dataset used in [Blok et al., 2020], a second dataset was collected in daylight without artificial illumination on a broccoli field in The Netherlands. There, an Intel Realsense D435 RGB-D camera mounted on a metal frame was used. Both the RGB and the depth image from this camera were registered to create one RGB-D image of 1280x720 pixels. Similarly, the RGB image and the depth image (from the stereo camera) in the USA dataset were registered, producing one RGB-D image of 1280x1024 pixels. In total, 947 and 1,613 RGB-D images of broccoli heads with various occlusion levels were collected from the USA and The Netherlands, respectively. However, a large portion of the occlusions were artificially created by placing loose leafs over the broccoli heads. All 2,560 RGB-D images from the two datasets were re-scaled and zero-padded to 1280x1280 pixels.

The size estimation algorithm first segmented the broccoli head in the RGB image using one of the CNN models. Then, the broccoli head diameter was estimated in the registered depth image using the mask produced by the CNN. On all test images, the Mask R-CNN achieved precision and recall scores of 98.4% and 97.3%, respectively; whereas the ORCNN achieved a precision of 97.6% and a recall of 97.8%. As for sizing, the ORCNN provided a better segmentation of occluded broccoli heads compared to Mask R-CNN achieving a lower mean absolute diameter error on the 487 broccoli heads counted in dataset. A detection example of an occluded broccoli head is shown in Figure 2.9. An interesting research avenue would be to test a similar sizing technique on segmentation methods working directly on 3D point clouds, as the CNN mask used on the registered depth image to estimate sizes proved to induce some errors. Additionally, a little more than half of the broccoli images had artificial leaf occlusions, implying a bias difficult to account for in the performance results.

[García-Manso et al., 2021] presented another system for localisation of broccoli heads based on a Faster R-CNN model built on a pre-trained ResNet-50 model using image sets from ImageNet. The original images used in this work were captured

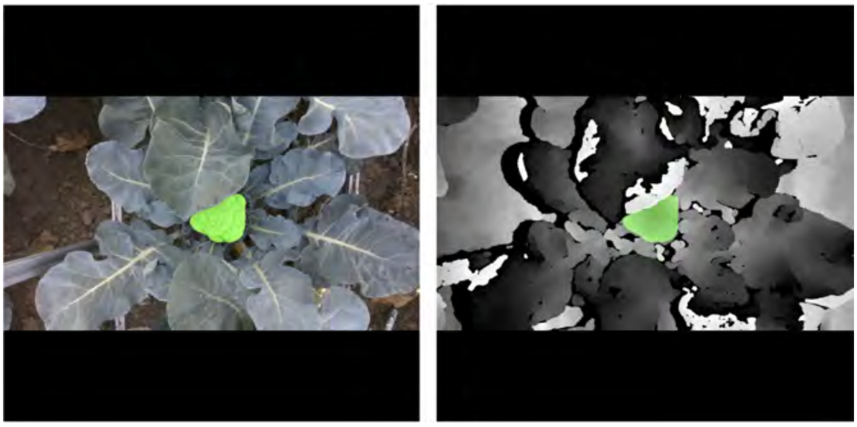


Figure 2.9: Example of the ORCNN segmentation. The left image shows the segmentation (green pixels) in the RGB image. On the right, a depth mask is created, by masking the ORCNN onto the registered depth image. Images reproduced from [Blok et al., 2021], best seen in colour.

with natural illumination in planted fields in Spain. The broccoli heads visible in the images were often partially occluded by leaves or covered with water droplets of dew. In total, 6,139 RGB images were captured with broccoli heads considered harvestable (*i.e.*, optimal maturity, determined by an expert), immature (*i.e.*, expected to grow more) and wasted (*i.e.*, not suitable for consumption due defects, diseases, or excessive ripeness).

One set of images was first captured by hand at a fixed distance from the ground in multiple sessions using two different cameras: 618 images at a resolution of 2592×1944 and 57 images at a resolution of 2288×1712. An additional set of 5,464 images was collected using a third camera placed on a tractor, automatically capturing over 10 images per second while the tractor was moving at a speed of approximately of 1 km/h. A total of 6,165 images downsized to 640x480 was produced. In this final set, 2,778 images of broccoli heads were considered harvestable, 2,805 immature and 582 wasted. Once taken, the images were marked (correct location of broccoli heads) and labelled as harvestable, immature and wasted by a human expert. The entire dataset was then randomly split into 80% for training and 20% for testing.

The system was able to correctly detect and classify 97% of the test images, including the ones partially occluded by leaves. Also, a mean average precision (mAP) score on all classes was reported to be of 83.5%. The mean processing time using an NVIDIA Jetson Nano computer was 109 ms per image. Figure 2.10 shows three

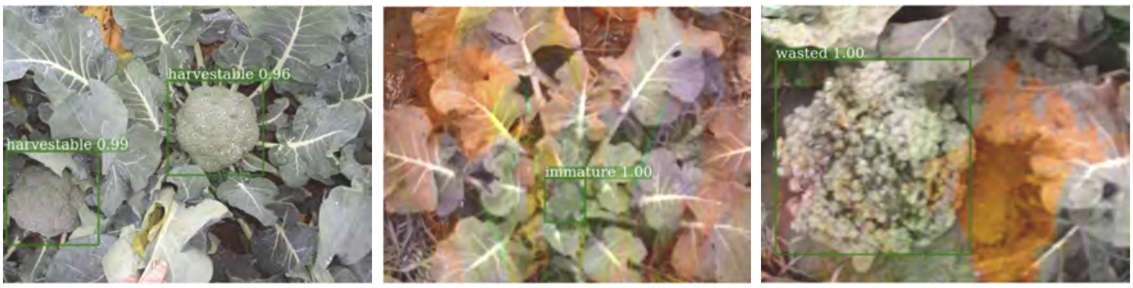


Figure 2.10: Examples of correct detections of the Faster R-CNN broccoli head detector. Two correctly detected harvestable broccoli heads on the same image on the left. Correct detection of immature broccoli head in the middle and wasted broccoli head on the right. The yellow colour are reflections produced by the sun in the image when they were captured. Images reproduced from [García-Manso et al., 2021], best seen in colour.

image cases correctly classified as harvestable, immature and wasted broccoli heads.

[Psiroukis et al., 2022] tested several deep learning architectures to detect and classify heads of organic broccoli based on their maturity level. The models were trained on RGB images captured from low-altitude UAV flights. The goal was of this study was to automate the process of human scouting required to initially identify the field segments where several broccoli heads have reached maturity. This scouting process is performed on foot to avoid the undesirable effect of soil compaction in organic farms caused by vehicles.

The aerial dataset was collected using a quadcopter drone equipped with an RGB camera. The captured images were of 4096×2160 resolution accompanied by the geo-referencing metadata of each image, collected at a fixed interval of 2 seconds during three different flights. Broccoli heads were annotated into three classes representing immature crops, nearly mature crops, and ready-to-harvest heads. Five different object detection architectures were tested, namely, Faster R-CNN and RetinaNet, both pre-trained on a ResNet-152 model, SSD pre-trained on MobileNet-V2, EfficientDet-D1, and CenterNet pre-trained on HG-104.

The results revealed that the object detection approach for automated maturity classification achieved comparable results to human scouting. Their respective performances were over 80% mAP@50 and 70% mAP@75 when using three levels of maturity, and even higher when simplifying the use case into a two-class problem, exceeding 91% and 83%, respectively. At the same time, geometrical transformations for data augmentations reported improvements, while colour distortions were counterproductive.

A wealth of other applications involving deep learning techniques in agriculture for different crops are also available in the literature [Liakos et al., 2018, Kamilaris and Prenafeta-Boldú, 2018, Yang and Xu, 2021].

There are several other applications where processing 3D information is essential in agriculture, such as harvesting fruits [Silwal et al., 2016, Silwal et al., 2017], weeding vegetable crops [Chen et al., 2018], crop phenotyping [Guo et al., 2018], monitoring of vineyards [Comba et al., 2018], and cane detection and localisation of soft fruits [Khanal et al., 2019]. All of them require 3D sensing for detection and localisation of crops.

2.3 Prototypes of broccoli harvesters

As seen so far, several methods able to distinguish between immature, mature, and post mature broccoli have been reported in the literature with varying levels of success. However, in limited cases the detection systems have been integrated with an actual harvesting device deployed in the field.

Given the complexity of the selective autonomous harvesting task, some early efforts started by developing purely mechanical harvesters of broccoli crops that performed both selective and mass harvesting [Walton and Casada, 1988, Casada et al., 1989, Shearer et al., 1990, Shearer et al., 1991b, Shearer et al., 1991a, Wilhoit and Vaughan, 1991]. These mechanical machines commonly included a cutting system and a set of conveyors to deposit the broccoli heads in towed wagons controlled from the operator's seat with various degrees of success. However, apart from the damage caused to the harvested heads and the necessary skills of the operator to drive and to select the crops to cut, one of the main drawbacks was that the harvesters were not limited by their mechanical capabilities but rather by the crop variety, as some cultivars have a majority of heads mature for a single harvest, but most varieties have less than 50% of heads mature during a conventional window harvesting.

More recently, other fully and semi-autonomous broccoli harvesters have been developed. Some of them have even passed several stages of field trials and are now commercially available. One of these machines was designed to mass harvest broccoli or cabbage using a picker belt system (with adjustable cutting high) synchronised with the travel speed of the harvester. It also included a camera system to monitor different areas of the machine [Univerco, 2021].

Other approaches opted for developing selective harvesters that consisted of devices attached to a conventional tractor, either fitted with blades and conveyor belts to later sort by hand the heads [Dobmac Agricultural Machinery, 2021], or with robotic arms that cut and move the broccoli crops into crates [FANUC UK Limited, 2019]. This last machine integrates a vision system that scans the planted field as the machine moves over the crops, locating the broccoli heads, assessing their size, and then sending coordinates to the robotic arm where the cutting tool is attached. The robotic arm then drops the heads into boxes of different sizes located at the end of the harvester, or into a loader trailer [RoboVeg Ltd, 2021].

Mechanisation and automation are widely used in agriculture, particularly in large farm operations. Nevertheless, a very small number of autonomous selective harvesters have evolved from prototype to fully operational machines, and none of them have seen a widespread use in open fields. This limited success is due to the complexity of the field itself, the environment, the desired harvesting task, and the cost involved in acquiring and operating such devices to make them economically feasible solutions.

2.4 Research Outline

Efficient and effective automatic detection of broccoli head remains a challenging problem in autonomous selective harvesting. This PhD research, focuses on advanced processing techniques on 3D depth information to allow real-time broccoli heads detection for autonomous robotic selective harvesting.

Harvesting is performed several times during the production of a high-value crop such as broccoli. Currently, a number of robotic harvesters prototypes for broccoli and other products exist (see Sec. 2.3), though they are not widely used due to its harvesting success rates and *cycle times* [Bac et al., 2014, Bachche, 2015, Kootstra et al., 2020], *i.e.*, the time of an average full harvest operation, including detection and localisation, ripeness assessment, cutting, transport of the cut crop, and robot/device movement to the next crop location. This time includes the time invested by both successful and failed harvesting attempts and it is important to determine the economic feasibility of the harvesting robot [Duckett et al., 2018, Kootstra et al., 2021, Vrochidou et al., 2022]. In addition, harvesting is one of the largest production costs in agriculture, as profitability depends on developing an efficient way to harvest a high quality produce. Therefore, developing methods to reduce the processing time in the various phases of the harvesting operation is highly desirable.

In particular, the detection system should be able to locate the crop given the high variability of both crop and its growing environment. Also, an effective grasping and cutting system is required to harvest the soft and delicate produce without damaging it regardless of their various shapes and sizes at higher precision and speed to improve productivity and lower the overall costs [Vrochidou et al., 2022].

Efficiently processing high frame rates is important towards these goals. Intuitively, processing sensory data at high frame rate should be always better, but is not really clear the most suitable frame rate for a particular application in agriculture. However, this entirely depends on the specific task and it could even be argued that a faster hardware also benefits faster processing times.

The ability to recognise how far away objects are in a scene is paramount in robotics applications. 3D perception allows robot to reach levels of autonomy not possible with 2D vision alone. In fully autonomous robotic applications in farm fields, 3D data allowed to considerably improve mapping and localisation algorithms, as 3D sensors collect rich information, including location, dimension, and orientation. That allows a robotic platform to reliably detect obstacles and segment graspable produce and other surfaces as well as the overall surrounding environment.

Over the past decade, 3D data has become widely available due to the introduction of new low cost sensors, such as the Intel Realsense cameras and the now discontinued MS Kinect, that provided depth information in addition to colour and infrared data. This sensors promoted a wealth of research on this new source of rich information for improving the cost and performance of many robotic perception tasks.

Even though 2D imagery has been the main focus in the literature so far for autonomous harvesting operations, this thesis focuses on effective and efficient 3D depth features under the hypothesis that they can provide better localisation, size estimation and other analysis related to shape useful for many applications in agriculture. Our primary goal is to design algorithms that reliably and efficiently extract segments of crops of broccoli plants. The methods presented in this dissertation capture and classify the sensed information based on local features of the extracted clusters of points by analysing the relationship of each point and the points in its vicinity.

This thesis extends current research results by detecting and tracking broccoli heads in real outdoor field conditions using affordable 3D sensors (*i.e.*, Kinnect and Realsense) in real time. The hypothesis evaluated in this research is that a robotic system using low-cost 3D sensors capable of detecting and tracking the target crop, improves harvest success and increases production while decreasing labour and other harvesting related costs. The overall goal of the thesis is to develop and to demonstrate machine learning algorithms capable of delivering high detection and tracking performance and real-time execution.

3 Real-time Feature-based Detection of Broccoli Heads

Agriculture is going through a constant evolution and technology is becoming a fundamental part of modern farms. The advent and proliferation of affordable and high resolution RGB-D sensors has enabled applications in many areas in agriculture, including inspection, quality control and, certainly, harvesting. Harvesting is one of the largest production expenses and profitability depends on developing efficient ways to perform harvesting operations of broccoli and other high value crops.

Different mechanical broccoli harvesters have been developed throughout the years to reduce human labour dependency and increase harvesting efficiency [Casada et al., 1989, Shearer et al., 1991b, Sarkar and Raheman, 2021, Dobmac Agricultural Machinery, 2021]. These mechanisation attempts have focused primarily on mass or slaughter harvesting because of simpler design requirements, lower hardware costs and higher harvest rates when compared with selective harvesting. Even though mechanical harvesters are able to enhance broccoli harvesting productivity, they lack the ability to differentiate the size and quality of the produce, which could severely damage broccoli heads with the consequent impact on their market value. Broccoli harvesting robots are expected to overcome the identified barriers of manual and mechanical harvesters [Zhang and Karkee, 2021].

One essential component of an autonomous selective robotic harvester is the recognition system to detect the produce in a 3D coordinate system, so the picking system can perform grasping and cutting operations to detach the crop from the plant. This recognition system involves both 2D colour and 3D sensors, as well as data processing techniques. Sensing and data processing methods for 3D is one of the critically important components as it plays a vital role in the harvesting robot's design because it affects its overall cost and operational speed [Karkee et al., 2021].

This chapter details four detection methods that operate at frame rates of up to 15 fps on datasets of crops of broccoli plants collected with affordable RGB-D sensors. These methods achieve high performance in detection by exploiting the organised structure of the point clouds being processed. We hypothesised that through well designed 3D features and the organised structure of point cloud data, conventional supervised learning models can efficiently detect and generalise on 3D data of multiple broccoli varieties. The goal of our study is to develop and demonstrate machine learning algorithms capable of delivering high classification performance and real-time execution.

The results of this chapter have been published in [Montes et al., 2019] and [Montes et al., 2020].

3.1 Materials and Methods

3.1.1 3D Point Cloud Datasets

This dissertation aims at providing reliable algorithms for the detection, segmentation and analysis of broccoli heads in 3D. The performance of the methods presented here have been evaluated using multiple datasets of crops of broccoli plants collected in real open field conditions with low cost RGB-D sensors in two different countries with complementary growing seasons. One dataset was recorded in locations in the United Kingdom, where broccoli is a summer crop, and another in Spain, where broccoli is a winter crop.

These datasets were first used and published by [Kusumam et al., 2016] and are available with no restricctions to the scientific community (see Section 3.1.1.1). In general, publicly available datasets are valuable as they reduce the data collection and preparation efforts and allow researchers to focus on the development of methods to improve detection rates. In this thesis, a number of corrections and enhancements have been made to these datasets, specially on annotations, as described in the following sections.

3.1.1.1 Data Acquisition

Autonomous selective robotic harvesting first must focus on collecting data to test the efficacy and efficiency of new developments. However, the process of gathering data in agriculture can be extremely time-consuming and open to human error [Lu and Young, 2020]. In the experiments presented in this dissertation, we use the datasets published by [Kusumam et al., 2017]. These datasets, collected with a RGB-D camera, are publicly available to enable results comparison and further development of methods for automated harvesting[1].

RGB-D cameras are sensing systems that simultaneously capture RGB images along with pixel-wise depth information. Depth sensors perform a dense per-pixel measurement of scene depth. These measured depth values are normally delivered as a 2.5-dimensional representation of the visible parts of the scene. An RGB-D sensor combines a conventional colour camera with one of such depth sensors. Modern RGB-D cameras allow to capture reasonably accurate resolution depth and colour information at high frame rates. Various RGB-D cameras have become widely available over the past decade due to the introduction of new low cost devices, such as the Intel Realsense cameras and the Microsoft Kinect.

The datasets from [Kusumam et al., 2017] were captured in planted farm fields in the UK and Spain under different weather conditions. The data was collected using a Kinect 2 RGB-D sensor which provides high resolution RGB images at 1920 × 1080 pixels along with a 512 × 424 depth resolution. The sensor was fixed inside a shrouded metal casing to protect the equipment from external conditions. Also, artificial illumination is fixed inside the casing to achieve uniform illumination.

The sensor was vertically mounted, pointing downwards at a height of 125-140 cm from the ground. The recorded RGB and depth data can readily be aligned and downsampled to the depth map resolution when captured without affecting frame rate speeds to form and deliver a single point cloud frame. The point cloud data was collected with the camera casing mounted on the rear of a farm tractor, as shown in Figure 3.1.

The UK set consisted of 600 frames of the broccoli variety *Ironman*, while the Spain

[1] https://lcas.lincoln.ac.uk/nextcloud/shared/agritech-datasets/broccoli/broccoli_datasets.html

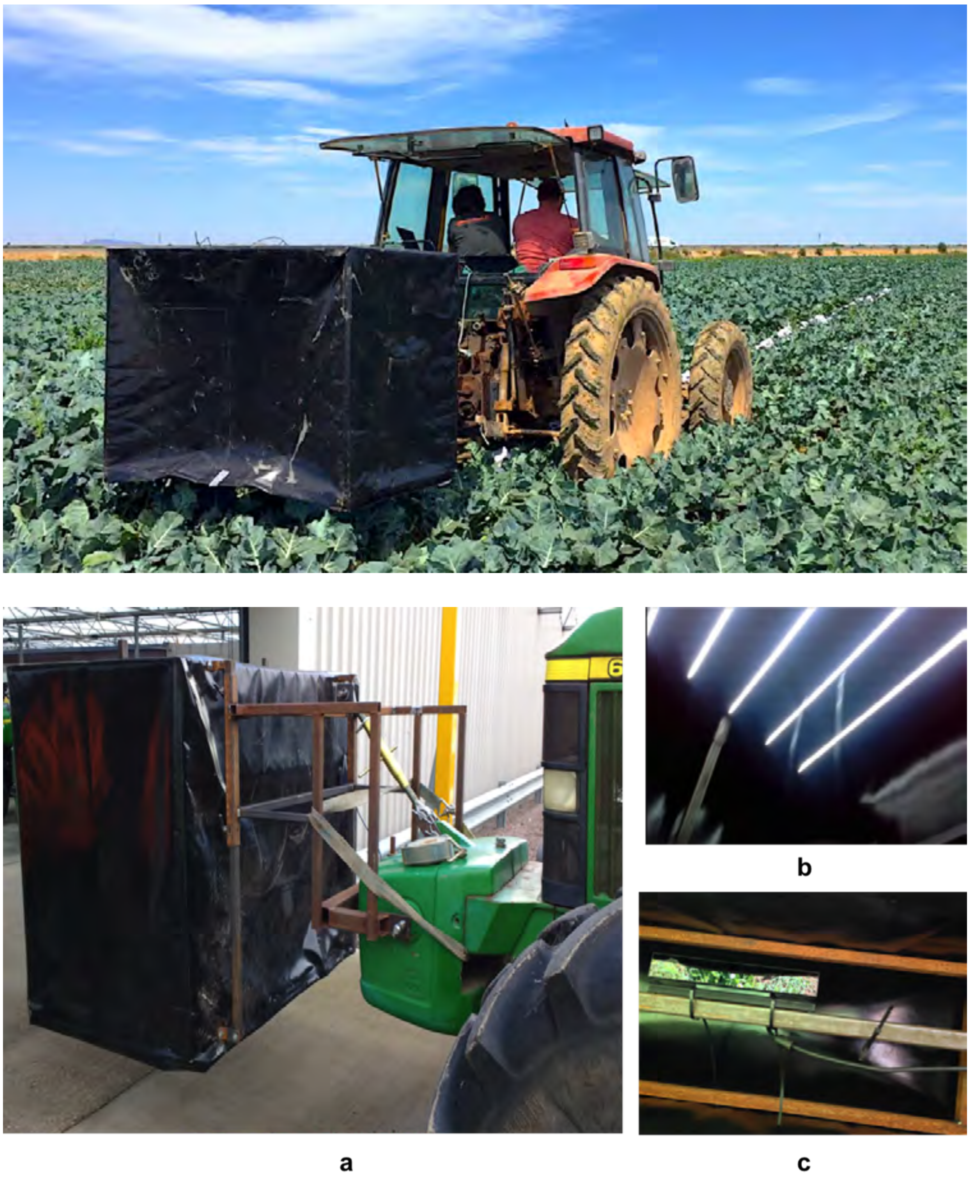


Figure 3.1: Data acquisition with the 3D sensor casing mounted at the rear of the tractor is shown in the top picture. In the pictures at the bottom, the sensor (c) is fixed inside a shrouded box (a) to block direct sunlight and to protect the equipment. An arrangement of artificial LED illumination is also fixed inside the box to achieve uniform illumination (b). Images reproduced from [Kusumam et al., 2017], best seen in colour.

set included 300 frames of the variety *Titanium*. 300 frames of the UK set (UK1) were captured at 7.5 fps with a frame overlap of 95% and the remaining 300 (UK2) were captured at 3.3 fps with 90% overlap, whilst the Spain set was acquired at 6.4 fps with 94% overlap. Samples of the datasets are shown in Figure 3.2.

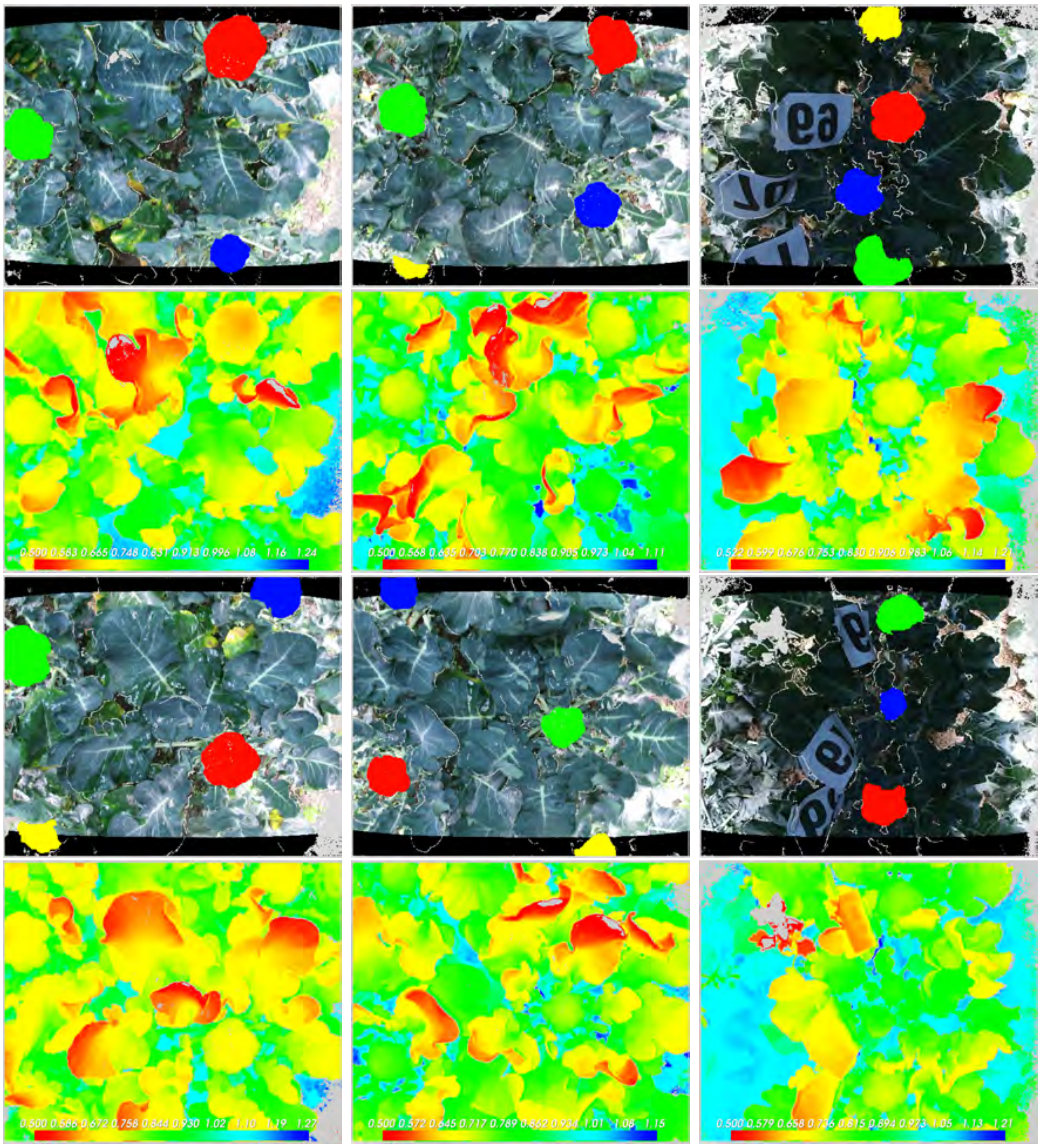


Figure 3.2: Frame samples of the UK dataset (first two columns, UK1 and UK2, respectively) and the Spain dataset (third column). The RGB-D camera captures colour (first and third rows) and depth features (second and fourth rows) that are joint together to create a single point cloud frame. Data of each frame is stored as a PCD (Point Cloud Data) file, *i.e.*, a file format for storing 3D point cloud data created as part of the PCL library [Cousins and Rusu, 2011]. One of the advantages of a PCD file is the ability to store and process organized point cloud datasets (see Section 3.1.1.3). Each point in the PCD format includes (x, y, z) coordinates and RGB colour data. The images (first and third rows) show, in different colours, the ground truth annotation points on the datasets. The 3D data on the black strips is also processed but no colour texture was matched by the sensor on those regions. Best seen in colour.

Dataset Name	Spain	UK1	UK2
Point cloud frames	300	300	300
Overlap Between Frames	94%	95%	90%
Average Width	1.10m	0.60m	0.60m
Average Height	0.86m	0.80m	0.80m
Average Distance	0.81	0.75	0.75
Broccoli Variety	Titanium	Ironman	Ironman

Table 3.1: Summary of the dataset characteristics. After RGB and depth data alignments, all frames in the datasets share the same resolution of 512 × 424 and are annotated following an instance segmentation format.

Different broccoli heads are visible in the two datasets. In the Spain set only one row of the crop is visible, whereas two rows appear in the UK set. The two black bands at the top and bottom of the point cloud frames are due to the alignment of the RGB frame to the depth frame, and the RGB edge distortion is also visible. Even if RGB information is missing on the edges, depth information and their aligned points are still present. Here the usage of 3D vision over 2D approaches is clear. Due to miss-alignment of the RGB to the depth information, processing it directly will lead to wrong localisation, shape and extraction of the objects [Blok et al., 2021, Kootstra et al., 2021].

The broccoli varieties in the two datasets are noticeably different in terms of shape, size and localisation, but share some common features. The average square distance from the nearest points to the Cartesian coordinate origin is 0.29 m for the UK set and 0.34 m for the Spain set. There is also a slight variation in orientation of the camera between the two sets. These differences produce points captured at different distances, with the consequent variation in broccoli head sizes and also make occlusions more evident.

The Spain dataset, even though smaller, offers a greater challenge as that cultivar grows with a larger canopy and more occluded heads. Also, borders of the point cloud present distortions due to the light and sensor, which affect crops found in such areas. A summary of the dataset characteristics is presented in Table 3.1.

3.1.1.2 Data Annotation

Capturing and annotating 3D data of real world agricultural environments is a very challenging task. Once the data has been collected, the contents of the datasets are merely points that need to be annotated in terms of the objects these points represent. Manually annotating this data is very costly because of the high annotation effort required to generate the amount of labeled data needed for training learning models. This often leads to a lack of widespread and readily available data in agriculture and is even considered as an obstacle towards the commercial feasibility of crop classification systems [Chebrolu et al., 2017, Lüling et al., 2021]

3D object detection using machine learning algorithms rely on a significant amount of manually labeled training datasets as ground truths. Often with RGB-D sensors, the RGB picture is annotated and the masks generated are then transposed to the aligned depth image. However in case of wrongly aligned frames, spatial information in the point cloud is wrongly annotated and misleading.

Since the original datasets from [Kusumam et al., 2017] only included annotation of the ground-truth centroid positions of broccoli heads on both the UK and Spain datasets, a considerable effort was placed into annotating these datasets on a per point basis. This per point annotations are useful for 3D broccoli head detection and necessary for 3D segmentation tasks. Therefore, to evaluate the algorithms presented in this thesis, the 3D points have been manually annotated directly in the point clouds on all datasets using a software tool specially built for this task. This helped to streamline the process of accurately labelling all broccoli heads points.

The software interface developed here leaves the control of an annotation session in the user's hands and offers basic functionalities, such as point or group selection. In consequence, for each frame in all datasets, an instance segmentation annotation is carried out, where each point has a class associated to it (background and broccoli), and each broccoli head has a different number to identify it in the point cloud. This makes all annotations, true 3D annotations. Screenshots examples of a point cloud frame is shown in Figure 3.3 as its points of several broccoli heads are been annotated.

Compared with annotating 2D images, acquiring and annotating large point cloud

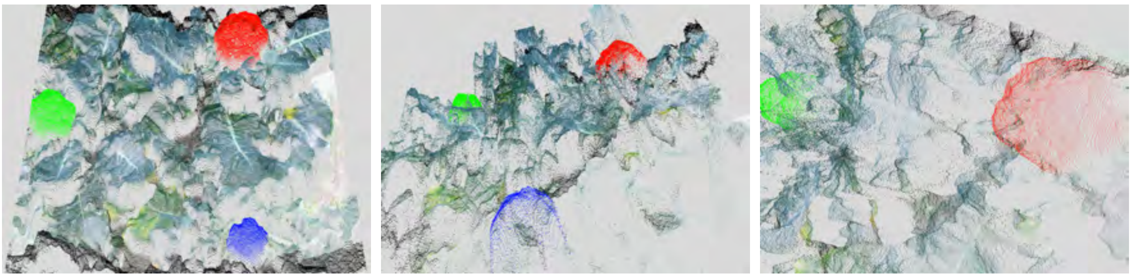


Figure 3.3: Screenshots of the software tool built for annotating the broccoli datasets. Each image shows a different viewpoint of the same point cloud frame, which can be moved, zoomed and rotated as needed to better select the broccoli head points. The points shown in red, green and blue colours are the ground truth annotations, which identify the same broccoli head, made by manually selecting groups of points which are in turn group together based on their proximity. Best seen in colour.

datasets is much more complicated. There is a wealth of software tools both freely and commercially available for annotating different types of Point Cloud data, particularly on datasets from both LIDAR and RADAR detection systems [OMahony et al., 2019, Merkle and Reiterer, 2022]. Most of these tools focus on bounding box annotations and some on object instance annotations using a variety of methods. Researchers have proposed both fully automated point cloud annotation techniques [Qi et al., 2018, Ku et al., 2018, Wang and Jia, 2019] and semiautomatic annotation tools [Castrejón et al., 2017, Acuna et al., 2018, Wang et al., 2019, Zimmer et al., 2019, Arief et al., 2020].

Even though other annotation tools are available developed for agricultural datasets [Miao et al., 2021, Le Louedec, 2021], for annotating and correcting the datasets used in the experiments presented in this thesis, we opted for a much simpler approach that allow us to annotate organized point clouds in PCD format using a simple yet useful and sufficient (and perhaps primitive) interface built with the tools provided by the PCL software library[2]. Such annotation task, albeit tedious and long, did not required the vast majority of features offered by other tools.

3.1.1.3 Organised Point Clouds

Low cost 3D sensors have emerged over the past decade to become one of the most versatile types of sensors used in robotics. For many applications, 3D sensing has

[2] https://pointclouds.org

become the *de facto* choice for tasks such as object detection, object inspection, and collision avoidance [Karkee et al., 2021]. Affordable 3D sensors are able to simultaneously capture high-resolution colour and depth images at high frame rates. These depth images can be converted into *organised point clouds*. An organised point cloud dataset is a cloud of 3D points that maintain a two-dimensional matrix layout structure, where the data is divided into rows and columns. Examples of such point clouds include data coming from Structured Light or Time-Of-Flight sensors, such as the Intel Realsense suite or the Microsoft Azure Kinect.

Structured light 3D sensors projects patterns of infrared light. The way the pattern deforms on the scene is then used to construct the depth map. Time-Of-Flight sensors, on the other hand, flood the scene with infrared light and calculate depth by the time it takes to return to the sensor. It is also possible to convert stereo images into organised point clouds provided that the camera's intrinsic calibration parameters are known.

The process of creating an organized point cloud involves correcting the image distortions caused by the lens of the sensor and a calibration matrix to estimate the correspondence between the RGB and IR (Infra Red) images. The distortions of the colour camera can be corrected using its intrinsic calibration parameters [Remondino and Fraser, 2006], whereas the correspondence can be done by a transformation (projective) matrix between the depth and colour streams [Rothwell et al., 1993]. The RGB image, usually bigger, is then cropped to fit the field of view of the IR image.

Organised point clouds can be useful in a wide variety of robotics applications. The main advantage of these point clouds is that the location of neighbouring points of any other point in the matrix can be more efficiently retrieved. This makes unnecessary to run costly searches and drastically speeds up processing times and lowers the costs of various 3D algorithms.

An added advantage of using a modern RGB-D camera is that 3D information can be readily acquired and processed into organised point clouds at frame rates of up to 30Hz, *i.e.*, the same frame rate delivered by the cameras used to collect the datasets of the experiments presented here. Therefore, no extra processing time needs to be invested in producing the organised data [Holz et al., 2011, Holzer et al., 2012]. This is done by fetching data from the RGB-D sensor via the OpenNI[3] drivers and then

[3] https://structure.io/openni

creating a PCD file using the PCL library.

In this thesis, this property of the datasets collected with a low cost 3D sensor is extensively used to process depth data at high frame rates.

3.1.2 Organised Broccoli Detection Pipeline

Two approaches are common in 3D object segmentation methods, namely, local and global object recognition pipelines. In the first approach, local methods commonly select a list of appropriate points, often referred to as *key-points*, and extract a set of features in the vicinity of those points. The points are then matched against a model and the correspondences are grouped according to the geometry of the model. In contrast, in a global recognition pipeline, the scene is segmented into smaller regions and global features are computed for each segment. These features are then matched with the descriptors of a model.

This section details a global recognition pipeline for real-time broccoli head detection. The pipeline includes four different clustering methods to group segments of 3D points. The entire process involves the following main steps:

1. a pre-processing stage which involves I) a filtering step based on a conditional removal of 3D points with a quadratic comparison, and II) a normal estimation step on organised point clouds,

2. a clustering stage which returns a list of point clusters with common attributes. Four different clustering strategies are considered in this stage: I) Fast Euclidean Clustering, II) Euclidean Angle Clustering, III) Organised Region Growing Segmentation, and IV) Organised Edge Segmentation,

3. a global 3D feature descriptor estimation step that uses the normal vector angles to encode cluster properties, and

4. a classification step for model learning and for predicting detections of broccoli heads.

These four stages of the detection pipeline are graphically depicted in Figure 3.4.

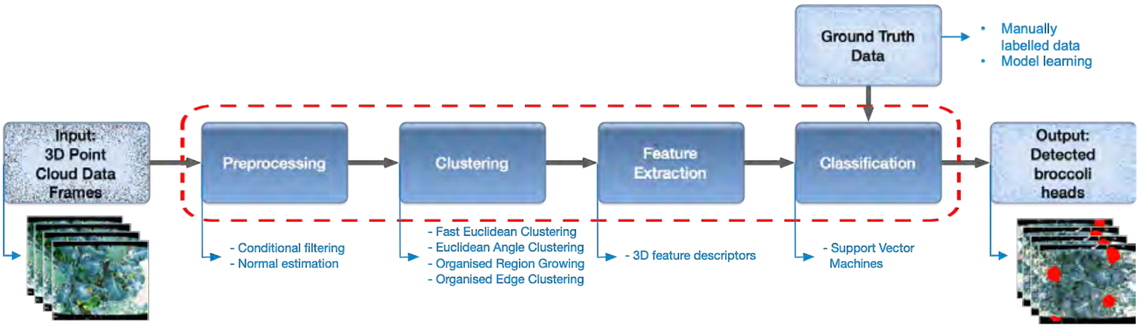

Ground Truth Data
Manually labelled data
Model learning
Input: 3D Point Cloud Data Frames
Preprocessing
Clustering
Feature Extraction
Classification
Output: Detected broccoli heads
- Conditional filtering
- Normal estimation
- Fast Euclidean Clustering
- Euclidean Angle Clustering
- Organised Region Growing
- Organised Edge Clustering
- 3D feature descriptors
- Support Vector Machines

Figure 3.4: Real-time 3D broccoli detection pipeline. The main steps of the pipeline are highlighted by a red bounding box. The input data are the point cloud frames acquired by the RGB-D sensor. The output is the locations of the broccoli heads detected on each input frame.

3.1.2.1 Pre-processing Stage

Quadratic Comparison Filtering

When sensing 3D data, errors and various other deviations are common to occur leading to noise in the point cloud data. This makes the estimation of some features, such as surface normals, more complicated. In some other instances it is useful to simply downsample the number of points involved in a particular operation to speed up processing times or to better handle the amount of sensed information. Either the size of the data or its intrinsic inaccuracies can cause significant errors in further processing. It is therefore advisable to remove or reduce some data with a suitable filter.

Removing outlier points based on statistical analysis has already been shown to be a costly process, whereas filtering out points beyond certain depth was a more practical approach [Kusumam et al., 2016]. However, because some distance variations from the sensor to the crop are present in the datasets, which can occur due to multiple factors, depth filtering needs to be constantly adjusted.

The approach used in this study is to remove points based on an alternative conditional criteria comparison. This criteria uses a quadratic XYZ comparison and examines whether the (x, y, z) components of a given point satisfy $p^T Ap+2v^T p+c < 0$. Here $p = (x, y, z)$ is a point vector of the cloud being examined; A is a 3×3 matrix, which defines the quadratic parts of a geometric shape; v is the 3×1 vector; and c is a scalar. Thus, by defining A as the identity matrix, $v = (0, 0, 0)$, and $c = 1$, the

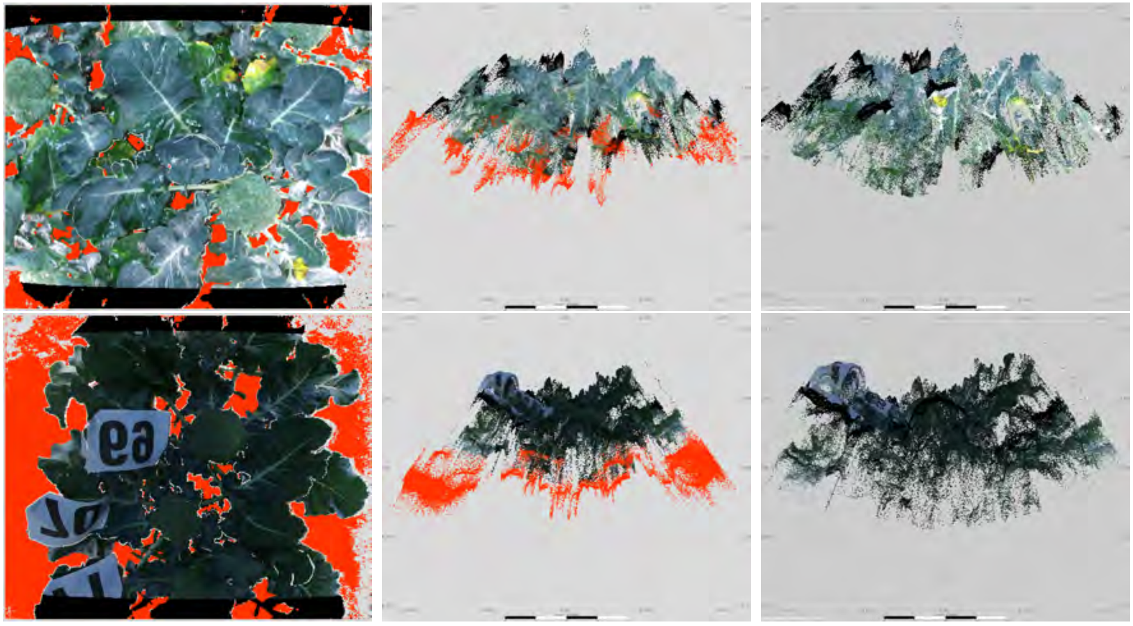


Figure 3.5: Frame samples of the UK dataset (first row) and the Spain dataset (second row) after being filtered by the quadratic XYZ comparison. The red areas are the filtered points shown from the sensor viewpoint and from a side view. To avoid breaking the organised structure of the point clouds, the filtered points are kept and set to a *NaN* value, which are ignored at processing time. Best seen in colour.

filter is applied using a spherical shape as follows:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}^T \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} + 2 \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}^T \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} + c = x^2 + y^2 + z^2 + c \quad (3.1)$$

This type of filter is particularly suitable for the type of sensor used in the datasets, as it removes points located at the greatest distance from the projected viewpoint of the sensor and preserves points in the areas of major interest. The effect of this filter can be seen in Figure 3.5.

Normal Estimation

Surface normals are important properties of a geometric surface. A surface normal can be considered a feature, although not a very discriminative one. Nonetheless, normals are important because they provide information about the curvature of the surface at some point, which can be used to compute more advanced features (such as the 3D feature descriptors detailed in Section 3.1.2.3) that encode properties of clusters extracted from the captured scenes of broccoli plants.

In 3D spaces, the normal of a surface plane at a point P is defined as the vector that is perpendicular to the plane that is tangent to the surface at P. As such, surface normals can also be calculated for the points of 3D point clouds. Normal estimation methods commonly work by either computing the tangent plane and the normal vector of a point as an average of the points within a given neighbourhood, or by fitting geometric primitives into the local neighbourhood of the current point. This could be a slow process, even for small point clouds. In general, normals are estimated without an specific direction, but by supposing that all vectors must point towards the RGB-D sensor they can all be re-oriented accordingly.

Integral images normal estimation Searching in a neighbourhood is commonly a slow process for many practical applications in robotics. Integral images provides a method for normal estimation on organised clouds [Holzer et al., 2012]. The algorithm uses the point cloud as a depth image while keeping a two-dimensional matrix layout of the data. This allows the algorithm to quickly create rectangular areas over which the normals are computed by taking advantage of the relationship between neighbouring points without the need of running costly searches. The result is a very efficient method to compute normal vectors using the inherent grid structure of the point clouds collected by modern low-cost RGB-D sensors. We use this algorithm to estimate normals for all the clustering methods detailed in Section 3.1.2.2. Figure 3.6 depicts an example of the normals computed for an organised point cloud and the associated normal map.

3.1.2.2 Clustering Algorithms

Efficient processing of point cloud data is central for building effective automated harvesting systems. It is therefore necessary to develop algorithms capable of delivering a high detection performance of crops. While deep learning architectures have shown favourable results, the amount of available labelled 3D data is often insufficient to achieve the generalisation needed in automated harvesting operations. This

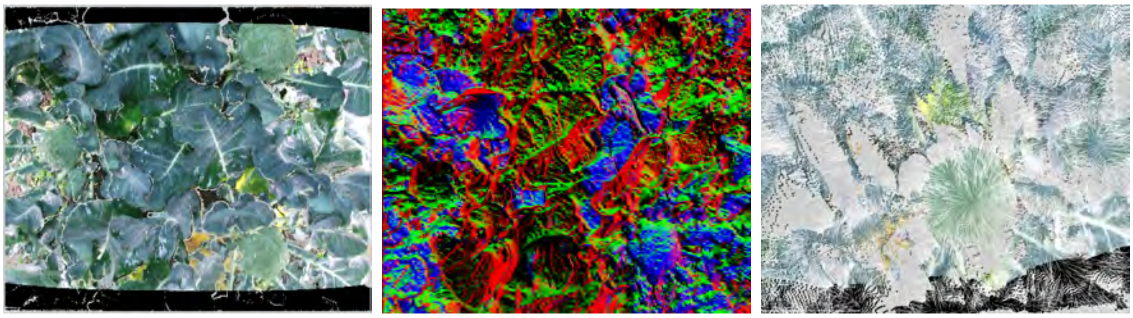


Figure 3.6: An example of an organised point cloud frame on the left and its associated normals map in the middle. In the normals map, the normal vectors are color-coded, *i.e.*, each vector component is represented by a different colour channel and then used here for visualization purposes only. On the right, a visualisation of the surface normals displayed as lines at every fifth point and zoomed in on one of the broccoli heads. Best seen in colour.

is where segmentation clustering techniques are relevant.

For an autonomous selective robotic harvesting system, vision is of the utmost importance to understand scenes of planted broccoli fields. To this end, algorithms are required to recognise crops and determine their location. The first step towards achieving this tasks is to segment the captured scenes, such that every segment represents a different object, either a broccoli head, surrounding leafs or patches of soil. A natural approach to tackle this segmentation problem is clustering points based on geometric and distance features. In this chapter, the problem of 3D scene segmentation is addressed by designing several clustering algorithms. The goal is to evaluate how one clustering algorithm may be more suitable than another for broccoli head segmentation based on the differences of each method to extract the clusters. An additional objective is to demonstrate how different clustering algorithms can be used to achieve better segmentation results.

The output of the clustering methods presented in this Section is a list of clusters. Each cluster groups points together based on a similarity criteria, *i.e.*, either because they are within a distance limit (FEC), or a normal angle threshold (EAC), or because they belong to the same area bounded by an edge (OES), or because the angle of their normal vectors and their curvature surface smoothness are within a similarity threshold (ORGS).

Regardless of the similarity criteria used, the clustering algorithm exploits the organised structure of the point clouds in the datasets and goes through the following steps: Firstly, a point is selected by the algorithm and added to the current cluster while marked as already processed. Then, it examines four neighbour points on the organised structure located on the left, right, top and bottom, and adds them to the cluster if they meet the similarity criteria. For every added point, its four neighbours are also checked until no more new points can be added. The cluster is then added to the list of clusters if it is within a predefined valid size. The algorithm then starts again with the remaining unprocessed points of the cloud. The steps of the clustering process are listed in Algorithm 3.1.

The clustering methods are now detailed in the following sections.

Fast Euclidean Clustering

Clustering algorithms allow data to be divided into subgroups, or clusters. Intuitively, these clusters share similar observations and therefore are highly dependent on how the notion of similarity is defined. The first and perhaps most intuitive similarity approach to clustering is based on the Euclidean distance between two points, *i.e.*, the further apart two points are in Euclidean space, the less similar they are, and *vice versa.* A Euclidean Clustering algorithm forms tight clusters for which two close points must belong to the same cluster. It also produces a list of clusters that are sufficiently far from each other.

A similar clustering method was adopted in the work published by [Kusumam et al., 2017]. The method implemented for this thesis, is a modified, yet faster version, of the Euclidean Clustering method, here dubbed Fast Euclidean Clustering (FEC). This method is efficient since it is based in the algorithm listed above (Algorithm 3.1), which makes extensive use of the organised structure of the point clouds to extract clusters of points. Examples of the clusters extracted by this method are depicted in Figure 3.7.

Euclidean Angle Clustering

A clustering algorithm relies heavily on the concept of similarity and subtle changes can have a positive effect on the output clusters. In the FEC method, an important number of clusters also include points that belong to other background elements, especially leaves that commonly grow around and close to the head of all broccoli cultivars. Because broccoli heads have a rounded tree-like shape, one feature that can be added to the similarity function of the algorithm is something that reflects

Algorithm 3.1 Organised clustering algorithm. The algorithm has been designed to take advantage of organised layout of the broccoli datasets used in this thesis.

```
function OrganisedClustering(PointCloud, SimilatyCriteria);
Input : A PointCloud and a SimilatyCriteria function
Output: A set of Clusters
Clusters = [] // list of clusters
processed = [PointCloud.size, false] // points processed
for point_ic in PointCloud do
    if processed[point_ic] then
        loop // point already processed
    CurrentCluster = []
    processed[point_ic]=true
    for point_cs in CurrentCluster do
        // left, right, top & bottom neighbouring points
        n1, n2, n3, n4 = get_four_neighbours(point_cs)
        if not processed[n1,n2,n3,n4]
            if Similarity(point_cs, [n1,n2,n3,n4]) then
                add PointCloud[n1,n2,n3,n4] to CurrentCluster
            processed[n1,n2,n3,n4] = true
    if $max\_size \geq CurrentCluster \geq min\_size$ then
        add CurrentCluster to Clusters
return Clusters
```

the natural shape of broccoli crops. Since surface normals estimation is part of the pipeline used in this study and normals provide information about the curvature of the surface at each point, the angle between them can be readily used as an added feature to reduce the number of points that are not part of a broccoli head.

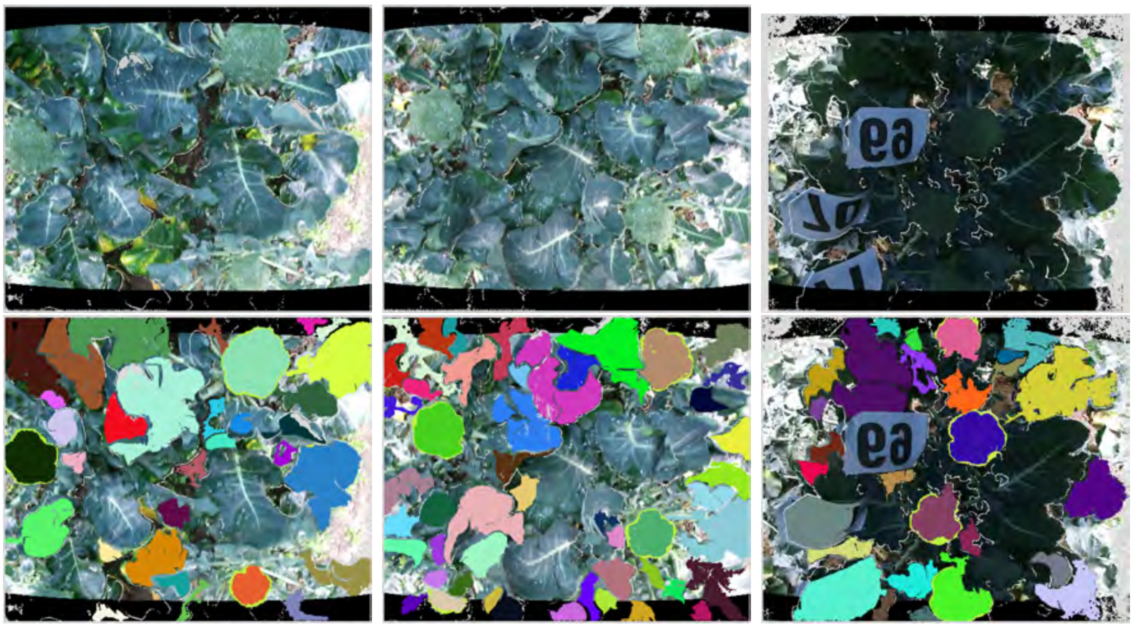


Figure 3.7: Examples of the output produced by the FEC algorithm on frames of both the UK and Spain datasets. Every coloured segment is one of the extracted clusters, *i.e.*, broccoli head, leaves or soil. Some clusters are overlapping points of ground truth annotations (shown in bright yellow). Best seen in colour.

Normal vectors of the broccoli heads points are oriented in different directions while forming different angles between them, as shown in Figure 3.6. The angle between two normal vectors is defined as the shortest angle at which any of the two normals is rotated about the other normal such that both of them have the same direction. In the Euclidean Angle Clustering method (EAC), if this angle is within a similarity threshold, in addition to a Euclidian distance threshold, then the point being compared is considered to be part of the same surface and is added to the current cluster. The angle θ between two normal vectors $\overrightarrow{n_1}$ and $\overrightarrow{n_2}$ is calculated as:

$$\theta = \arccos\left(\frac{\overrightarrow{n_1}\cdot\overrightarrow{n_1}}{\overrightarrow{|n_1||n_2|}}\right) \tag{3.2}$$

Figure 3.8 shows examples of the clusters extracted by this method.

Organised Region Growing Segmentation

In the FEC algorithm, neighbouring points are part of the same cluster if they are within a predefined Euclidean distance threshold. The EAC algorithm allows point with this same distance criteria to be kept if the angle of their normal vectors is

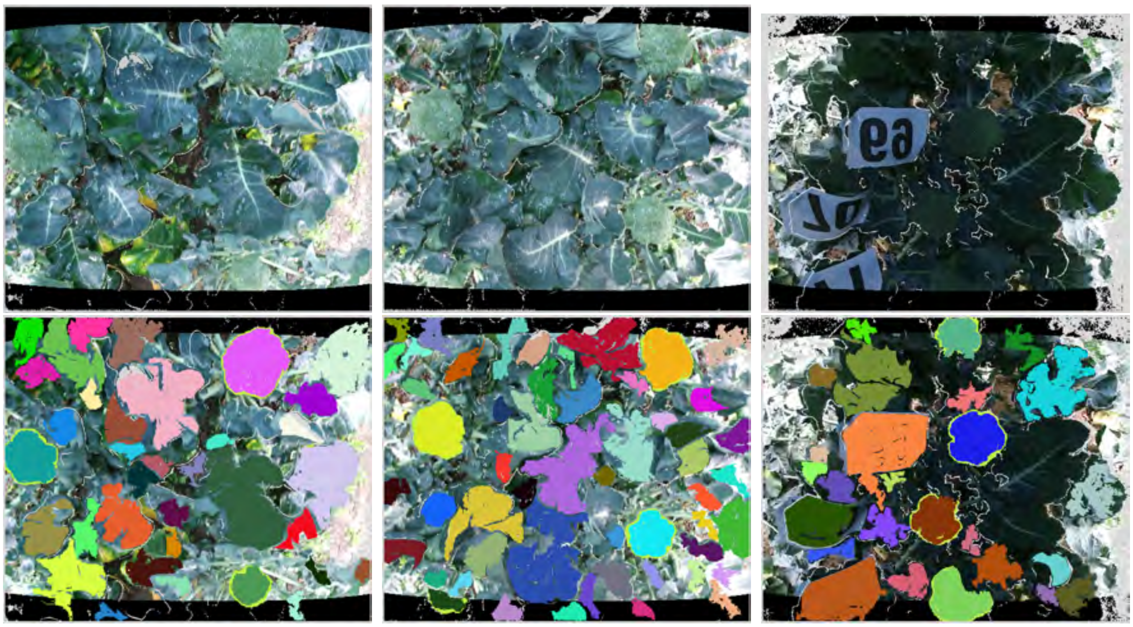


Figure 3.8: Examples of the output produced by the EAC algorithm on frames of both the UK and Spain datasets. The coloured segments are the set of clusters produced by the method. Best seen in colour.

also within a predefined threshold. According to Algorithm 3.1, every point already added to the current cluster is examined so that its neighbouring points can also be added and, in turn, further examined to add its neighbours until no more points meet the similarity criteria. However, there are always some discontinuities in the point cloud data that make some distant points that belong to a broccoli head not to be added to the correct cluster. Conversely, other points that are part of a background element are often added because they are too close to a broccoli head. An alternative is to make a cluster representing a region to grow by inspecting the surface curvature around each point.

The surface curvature at a point p can be defined as the amount of change in direction of its surface normal. The surface curvature can be estimated as a relationship between the eigenvalues of the covariance matrix $\mathcal{C}$ created from the nearest neighbours of a point p, as follows:

$$\mathcal{C} = \frac{1}{k}\sum_{i=1}^{k}(p_i - \overline{p}) \cdot (p_i - \overline{p})^T, \mathcal{C} \cdot \overrightarrow{v_j} = \lambda_j \cdot \overrightarrow{v_j}, j \in \{0, 1, 2\} \tag{3.3}$$

Where k is the number of point in the neighbourhood of p_i, $\overline{p}$ represents the 3D

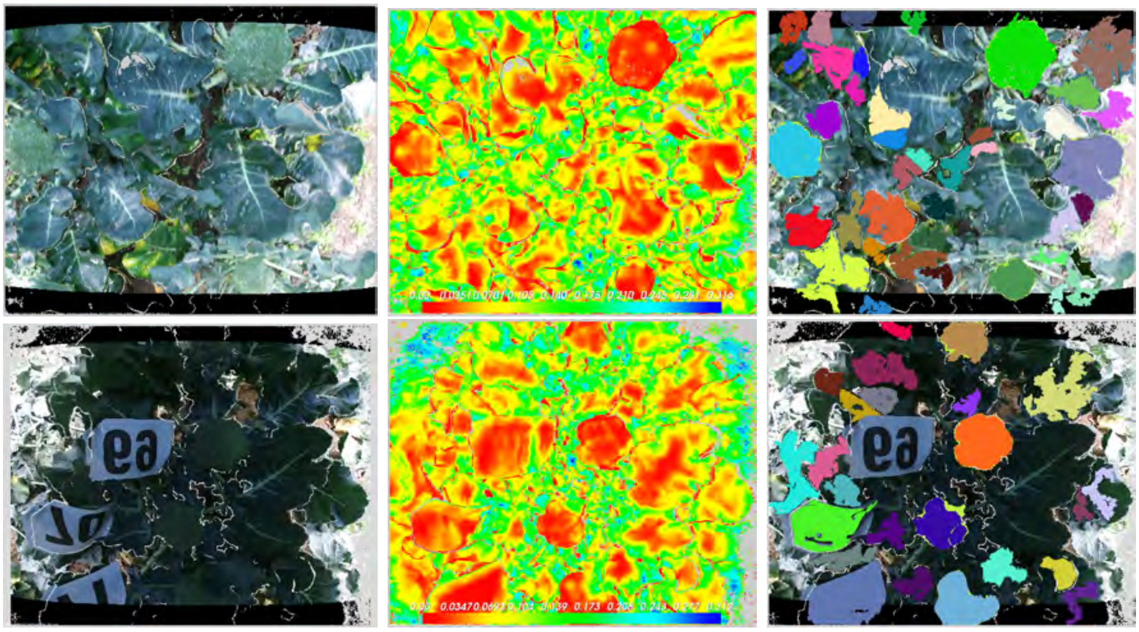


Figure 3.9: Examples of the output produced by the ORGS algorithm on frames of both the UK and Spain datasets. The original point cloud is on the left and its associated surface curvature map is in the middle. The red colour indicates regions of low curvature and the green-blue colour indicates regions of high curvature. On the far right, the coloured segments are the set of clusters produced by the method. Best seen in colour.

centroid of the nearest neighbours, λ_j is the j-th eigenvalue of the covariance matrix, and $\overrightarrow{v_j}$ the j-th eigenvector. The eigenvalues λ_j are then used as approximations of the surface curvature variation around p. If $\lambda_0 = min(\lambda_j)$, the curvature variation σ_p of a point p along the surface normal can be estimated as:

$$\sigma_p = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \tag{3.4}$$

In the Organised Region Growing Segmentation (ORGS) algorithm, neighbouring points are part of the same cluster if the angle of their normal vectors is within a predefined threshold. Then, some of the points added to the current cluster are used to make the region grow if their surface curvature, as defined above, is also within a threshold value. The idea is to make the region spread to other points in the vicinity with similar surface curvature. Points of high curvature do not make the region grow, as those point represent an abrupt surface change. Examples of the clusters extracted by this method are shown in Figure 3.9.

Organised Edge Segmentation

Features such as corners, edges, colour, and key-points have been widely used in 2D robotic perception for object recognition applications. These features can also been used to process RGB-D information. Our broccoli point cloud datasets contain depth points that were aligned with the RGB texture data, but also points for which no colour texture was matched by the sensor. For this reason, 3D edge features are of particular interest because they are applicable in both textured and textureless data.

The 3D edge detection algorithm, introduced by Choi *et al.* [Choi et al., 2013], labels points as edges based on point depth discontinuities and high curvature regions to search for salient geometric edges. It uses a tolerance distance to determine the difference in depth values between neighbouring points. All points within a chosen neighbourhood are readily accessible using the organised layout of the point cloud to efficiently detect edges. Similarly, the algorithm uses a neighbourhood size to label points as one of the predefined types of edges: *occluding* and *occluded* edges in this algorithm.

The method, called Organised Edge Segmentation (OES), first computes a set of edge labels that corresponds to all points in the point cloud. After detecting edges and labelling the points accordingly, the point cloud is then processed to extract clusters of points surrounded by the same edge. The procedure works by grouping points together that are not part of an edge and spreads to other points in the immediate vicinity to form clusters as outlined in Algorithm 3.1. Figure 3.10 shows the set of points that have been labelled as edges on two frame examples.

Clustering algorithms allow to extract a list of segments that, in turn, become the necessary input for other processes of a detection pipeline, such as feature descriptor estimation and classification. These two processes are introduced in the next sections.

3.1.2.3 3D Feature Descriptors Estimation

Among the different processing steps involved in 3D broccoli head detection, the design of feature descriptors is crucial, as their encoding properties are key on the overall detection results. The goal of the feature descriptor is to capture the under-

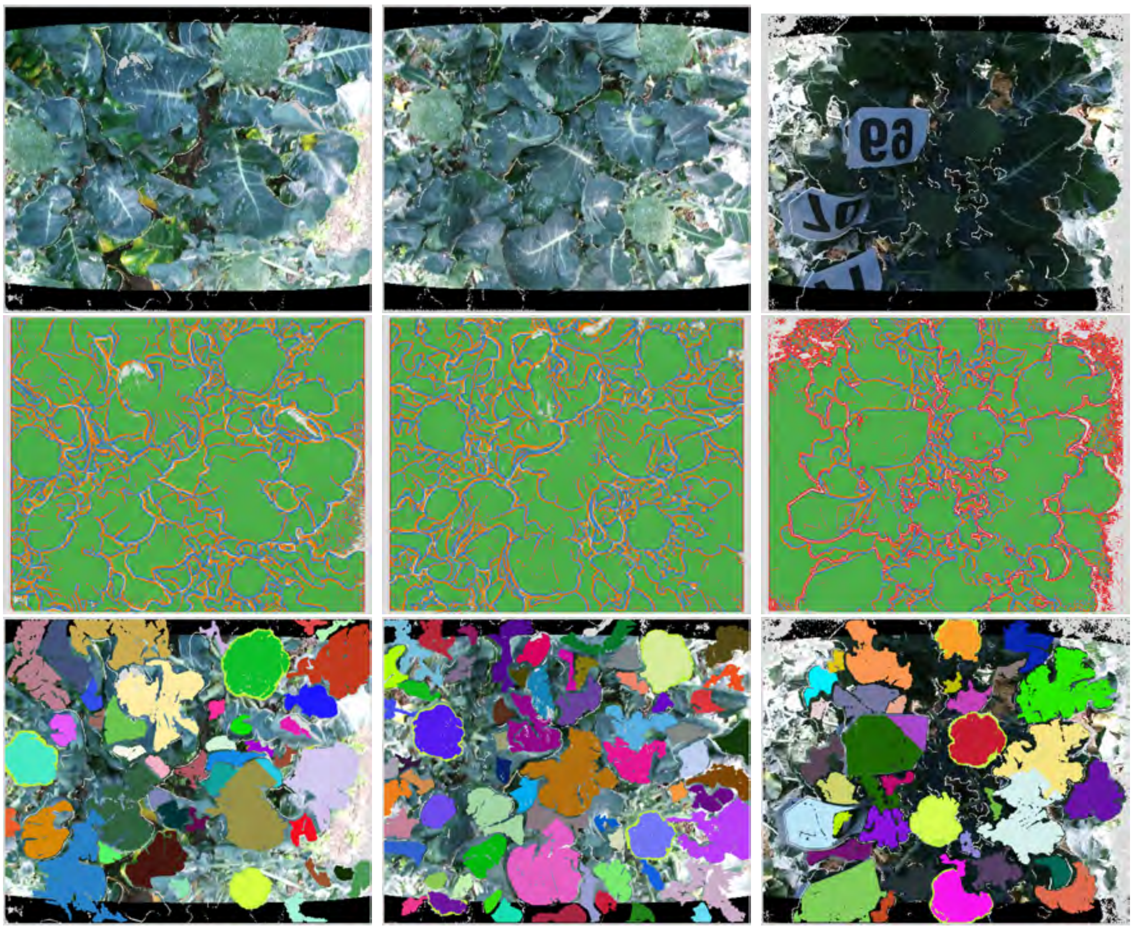


Figure 3.10: Examples of the edge detector produced by the OES algorithm on frames of both the UK and Spain datasets. In the middle row, the different types of edges are coloured in blue and red. On the bottom row, the coloured segments are the set of clusters produced by the method. Best seen in colour.

lying structure of the extracted clusters.

The various techniques adopted for object description can be divided into two main categories: global or local. While local descriptors are a more common choice for object recognition within cluttered scenes, they remain less discriminating due to the limited nature of their local scope [Rusu et al., 2009]. Global descriptors, on the other hand, can better capture relationships between sets of more distant points, while remaining robust to clutter and occlusion [Grilli et al., 2017].

A 3D feature descriptor should encode the surface properties of clusters and thus be able to discriminate between broccoli heads and other background clusters. In this chapter, the feature descriptor used in the broccoli detection pipeline is the

Viewpoint Feature Histogram (VFH) descriptor [Rusu et al., 2010]. The VFH is a global 3D feature descriptor that uses the distribution of the normal vector angles to represent the properties of data points within the same cluster. Also, the VFH descriptor can capture more robustly the structure of the objects by relying on low-level features, such as point surface normals.

The VFH descriptor works as follows: first, the pan, tilt and yaw angles between every point's surface normal and the centroid of a cluster are computed, *i.e.*, for a pair of a 3D point p and the centroid c of its cluster, and their corresponding estimated surface normals n_p and n_c, the set of normal angular deviations, can be estimated as:

$$\alpha = v \cdot n_c \tag{3.5}$$

$$\phi = u \cdot \frac{c - p}{d} \tag{3.6}$$

$$\theta = \arctan\left(w \cdot n_c, u \cdot n_c\right) \tag{3.7}$$

where $d = \|c - p\|$ and v, u, w represent a *Darboux* frame coordinate system (*i.e.*, a moving frame of reference constructed on a surface) chosen at p. After computing the sets of $(\alpha, \varphi, \theta)$ between all pairs of points $p_i, i \in \{1 \ldots n\}$ and the cluster centroid c, a *viewpoint* component is added to the descriptor.

The viewpoint component is calculated using the set of the angles that each normal makes with the viewpoint direction, *i.e*, the angle between the central viewpoint direction translated to each normal. This component measures the relative pan, tilt and yaw angles between the viewpoint direction at the central point and each of the normals on the surface. Then, the viewpoint angles are divided into 128 bins and the $(\alpha, \varphi, \theta)$ angles into 45 bins each for a total of 263 dimensions of the VFH feature descriptor. To make the VFH invariant to scale, the bins can be normalised using the number of points in each cluster.

The VFH descriptor transforms individual 3D point characteristics into cluster features useful for model learning and for distinguishing one cluster from another.

3.1.2.4 Model Learning and Classification

The last stage of the broccoli detection pipeline is model learning and classification. At this point, an object detection pipeline commonly relies on a robust classification algorithm, as the choice of the right classifier is of paramount importance because it might improve or worsen the detection results. In this dissertation, supervised machine learning algorithms (*i.e.*, classifiers) are used to learn cluster features of the captured 3D broccoli head points.

The experimental results presented here, use models learnt by the classifier to distinguish between broccoli heads and other background objects. For training the classifier, we use a set of training data consisting of n input feature vectors $x_1, x_2, ..., x_n$, where $x_i \in \mathbb{R}^n$ (computed as described in Section 3.1.2.2), along with their corresponding class labels $t_c, c \in \{0, 1\}$. The purpose of the classification algorithm is to produce a model which predicts one of the target classes values t_1 (broccoli head) or t_2 (other background element) for each feature vector x. All classification results are achieved using Support Vector Machines (SVM).

SVM are a popular machine learning method for classification, regression, and other learning tasks [Boser et al., 1992]. The objective of the SMV algorithm is to find a *hyperplane* in an n-dimensional space of features that classifies the set of descriptors. However, there are many possible hyperplanes that could be chosen to separate the two classes of descriptors considered in this study. Thus, the first task of the SVM algorithm is to find a plane that has the maximum distance between descriptors of both classes. This distance is called *margin* and maximising its value provides some additional gap size, so that feature vectors can be classified with higher confidence. Support vectors (hence the name) are descriptors that are closer to the hyperplane, and the margin of the classifier is maximised using these vectors. The hyperplane is considered a decision boundary, so descriptors falling on either side of the hyperplane can be labelled with one class or the other.

3.2 Evaluation and Results

The average number of clusters extracted per frame using the clustering methods detailed in Section 3.1.2.2 is 48 for the UK set and 54 for the Spain set. This implies a highly imbalanced class distribution between positive (*i.e.*, broccoli heads, 8.3%

UK, 5.6% Spain) and negative (*i.e.*, leaves, soil, etc.) samples. The challenge is to test the different clustering schemes to evaluate the generalisation performance and time execution of the broccoli detection pipeline. These results will also help to decide hardware configurations for an autonomous robotic harvester.

3.2.1 SVM Training

To train SVM for classification problems, the values for some parameters must be specified, namely, the regularisation cost parameter C and the *kernel* function (along with its corresponding parameters). In SVM, the parameter C controls the trade-off between training errors and generalisation or complexity of the classifier, while kernels help to deal with high dimensional classification spaces. SVM are shown to be efficient even in cases where the data is not linearly separable. It can also be used to classify data in higher dimensions using kernels. In this sense, RBF (radial basis function) kernel is a common and effective choice for binary classification [Hastie et al., 2004]. RBF kernels are the most generalised form of kernelisation and is one of the most widely used kernels due to its similarity to the Gaussian distribution [Burges, 1998]. The RBF kernel function for two points x and x' computes the similarity or how close they are to each other. This kernel can be represented as: $K(x, x\prime) = exp(-\gamma \cdot ||x - x\prime||^2)$, where γ is the kernel scale parameter, and $||x - x\prime||$ is the Euclidean (L$_2$-norm) distance between the two data points x and x' under scrutiny.

Finding the right γ along with the value of C is important and can be tuned for the given data by using hyper-parameter tuning techniques like Grid Search cross-validation and Random Search cross-validation. For each parameter setting, SVM obtains cross-validation accuracy and the parameters with the highest accuracy are returned. All classification results were performed using the following parameters: $C = 0.57665$, $\gamma = 0.410847$ for the RBF kernel, and class weights w adjusted inversely proportional to class frequencies in the input data as $w_j = n/(c \times n_j)$, where n is the total number of observations; n_j is the number of observations in class j, and c is the number of classes. These parameters were determined by *k-fold* cross validation ($k = 5$) based on a grid search.

The classifier training dataset consisted of randomly selected point cloud frames and all their extracted broccoli and background clusters. For experiments on the same

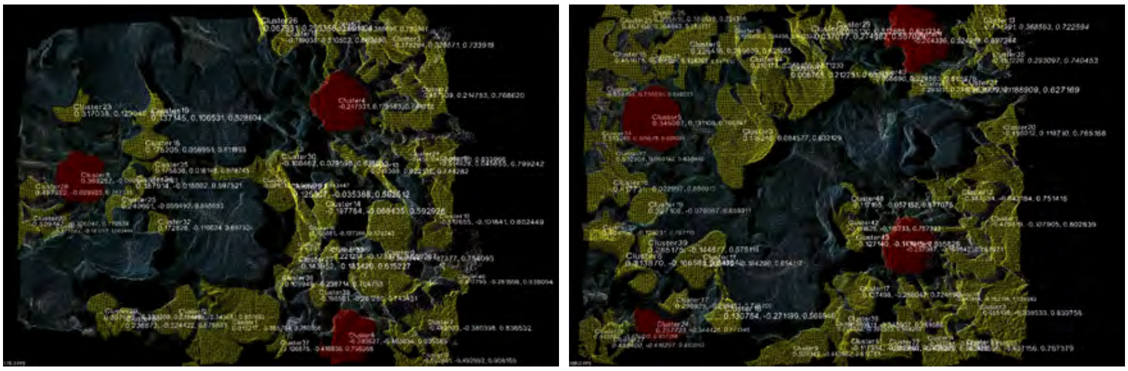


Figure 3.11: Examples of the output produced by the SVM classifier on frames of the UK dataset. The clusters labelled as background class are shown in yellow, whereas the ones labelled as broccoli head are shown in red. All clusters show a unique identification number, along with the (x, y, z) positions of their centroid locations. Best seen in colour.

set, a proportion of 75% frames with the annotated data were used for training and the remaining 25% were processed by the algorithms and used for testing. In any other case, 100% of one dataset was used for training and 100% of the other set was used for testing.

For each cluster generated by the pipeline, a corresponding VFH feature descriptor is produced. These descriptors, in turn, form the set of training and testing samples to be classified. Using SVM, the final classification output of the pipeline is a set of clusters representing the locations of broccoli heads, as shown in Figure 3.11.

All experiments have been implemented using the algorithms available as part of the PCL library [Cousins and Rusu, 2011] for processing point clouds, and the SVM implementation from the machine learning module included in the OpenCV library [Pulli et al., 2012].

3.2.2 Evaluation Metrics for the Broccoli Head Detection Pipeline

The performance of the clustering algorithms and the detection pipeline were evaluated by two different sets of metrics. The first one measured the individual broccoli head detection via the classifier confidence level and the intersection over union (IoU) metric using the Precision, Recall and F1 Score. The second evaluation metrics used were the Precision-Recall Curve (PRC) and the mean Average Precision (mAP).

3.2.2.1 Precision, Recall and F1 Score Evaluation

In the first evaluation, the detection performance was measured via the Precision, Recall and F1 Score metrics using the confidence level value returned by the SVM classifier and the IoU metric.

SVM and other machine learning models provide information about the reliability of their predictions by assigning a score value to how confident the model is about its response, either during training or at inference time. This score represents a confidence level commonly returned as an ordered set of values for all predictions made. Because a correct detection means that our algorithms should cluster the majority of the points within a broccoli head, the detection success was also evaluated by measuring the overlap between each positive prediction and its expected truth outcome. This overlap was calculated via the IoU metric. IoU (also known as the Jaccard index) is a metric for quantifying overlap ratio between a prediction and ground truth data. IoU is a score with values within the range zero (the lowest possible value, meaning no overlap) and one (the highest possible value, indicating full or perfect overlap). The formal definition of IoU is given as follows:

$$IoU(P, G) = \frac{\mid P \cap G \mid}{\mid P \cup G \mid} \tag{3.8}$$

where P is the area of the prediction and G is the area of the ground truth annotation.

A confidence score greater than a threshold value determines whether there was a correct broccoli detection, while an IoU value greater than a threshold determines whether there was enough overlap between the detected broccoli cluster and ground truth annotations. These two thresholds determined the number of true positives (TP), false positives (FP) and false negatives (FN) detections.

A TP occurs when both the confidence score and the IoU are greater than their corresponding thresholds (*i.e.*, a cluster detected as broccoli sufficiently overlapped the points labelled as broccoli). A FP detection is determined when the confidence score is greater than its threshold but the IoU value is less than the threshold value (*i.e.*, a background cluster detected as broccoli head or not enough overlap exists for a positive broccoli detection).

Similarly, either a confidence score or a IoU value less than their corresponding thresholds means a FN detection (*i.e.*, a broccoli for which a cluster was not extracted). The ratio of TP, FP and FN determine the Precision (Pr) and the Recall (Rc) computed as:

$$Pr = \frac{TP}{TP + FP} \tag{3.9}$$

$$Rc = \frac{TP}{TP + FN} \tag{3.10}$$

Precision is the percentage of correct detections, while Recall measures how well the system pipeline is able to cluster and predict all the annotated broccoli points. The F1 Score combines the precision and recall ratios into a single metric by computing their harmonic mean as follows:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3.11}$$

Table 3.2 summarises a list of evaluation results of the clustering algorithms used in the detection pipeline.

The scores in Table 3.2 for each dataset combination indicate an improved performance from recent published results, such as [Blok et al., 2016], [Kusumam et al., 2017] (using the same datasets). These results are also comparable with the most recent results based on deep learning techniques, albeit less general throughout all datasets. Nevertheless, because Precision, Recall an F1 scores are determined by the number of FP and FN, these two values can be more properly assessed in the automated selective harvester operation, as they largely correspond to small or partially captured broccoli heads. The former do not need to be harvested until fully grown and the latter are mostly seen at the frames edges and will be detected in posterior frames. Moreover, a low Precision and a high FN would eventually cause unnecessary actions by the broccoli cutting system.

	Precision	Recall	F1
OES UK v UK	98.6%	**96.7%**	**97.6%**
OES UK v Spain	**98.9%**	86.8%	92.5%
OES Spain v Spain	98.4%	93.8%	96.0%
OES Spain v UK	96.4%	88.7%	92.4%
FEC UK v UK	**98.4%**	**97.6%**	**97.6%**
FEC UK v Spain	91.4%	83.9%	87.5%
FEC Spain v Spain	96.0%	89.6%	92.7%
FEC Spain v UK	98.1%	91.3%	94.6%
EAC UK v UK	**98.4%**	**92.5%**	**95.4%**
EAC UK v Spain	96.1%	81.6%	88.3%
EAC Spain v Spain	97.8%	92.1%	94.9%
EAC Spain v UK	97.4%	86.0%	91.3%
ORGS UK v UK	**99.2%**	**91.7%**	**95.3%**
ORGS UK v Spain	94.8%	86.1%	90.2%
ORGS Spain v Spain	98.3%	93.4%	95.8%
ORGS Spain v UK	95.1%	90.1%	92.5%
[Blok et al., 2016]	**99.5%**	91.2%	95.2%
[Zhou et al., 2020]	89.6%	87.9%	88.7%
[Blok et al., 2020] Mask R-CNN	98.7%	93.9%	96.2%
[Blok et al., 2021] Mask R-CNN	98.4%	97.3%	**97.8%**
[Blok et al., 2021] ORCNN	97.6%	**97.8%**	97.7%

Table 3.2: Summary of *Precision*, *Recall* and *F1* evaluation metrics for all UK and Spain datasets combinations for each clustering method. UK is a combined set of all 600 RGB-D frames from the UK1 and UK2 sets. The list also includes, at the bottom rows, the most recent results published in the literature on broccoli detection .

Because broccoli heads grow and mature at different rates, in current practice, they are harvested by teams of broccoli pickers in no more than two passes to make the task economically viable. A robotic harvester, however, offers multiple opportunities for detection, as small heads will grow to the desired size and cab be harvested in another field pass.

Even when the above argument is given from the harvester practical point of view, detection methods should always strive for the highest measurable performance. Although the results of Table 3.2 show a high Precision score for every detection outcome, Recall is not equally high on every result showing some generalisation shortcomings on those algorithms. That is more evident for experiments where the UK and the Spain datasets are combined. One reason for these discrepancies is the notable difference between the two sets, particularly in the distance from the sensor to the broccoli plants and the high number of occlusions visible in the Spain set, as detailed in Section 3.1.1.1.

With a single set of thresholds for both the classifier confidence score and the IoU metric, Precision and Recall may not express whether the detection results precisely located the broccoli heads and robustly handled the imbalanced class distribution of the broccoli datasets. The Precision Recall Curve and the mean Average Precision provide more accurate measurements when both these thresholds fluctuate. These are the two metrics used in the next section.

3.2.2.2 PRC and mAP Evaluation

In the second evaluation, the detection performance was measured using the *Precision-Recall Curve* (PRC) and the mean *Average Precision* (mAP).

The PRC is a useful measure of prediction success as it has been shown to provide a more accurate interpretation of a classifier performance when the class samples are highly imbalanced [Saito and Rehmsmeier, 2015]. The PRC shows a trade-off between Precision and Recall for different classifier confidence score thresholds. Precision is the ratio of correct broccoli detections (*i.e.*, a sample predicted as broccoli head has indeed been labelled as such), whilst Recall measures the number of truly labelled broccoli heads detected.

A high area under the PRC represents both high Precision (*i.e.*, a low FP rate) and high Recall (*i.e.*, a low FN rate), while high scores for both indicate that the classifier responses are accurate (*i.e.*, high Precision) and that it is correctly predicting the majority of all truth results (*i.e.*, high Recall).

However, Precision may not decrease with Recall when the classifier confidence threshold varies. Lowering the threshold may increase the number of TP, which will increase Precision. Lowering the confidence threshold even more will introduce more FP, which will decrease Precision. Meanwhile, Recall may remain unchanged if the confidence threshold varies, as lowering the threshold may increase Recall, by increasing TP. Further lowering the threshold may leave Recall the same, while

Precision changes.

The PRC shows the balance between the Precision and Recall scores, as a high area under the curve represents high values for both metrics. This trade-off between Precision and Recall can be observed in the PRC, as a small change in the classifier confidence threshold considerably reduces Precision, with only a small increment in Recall.

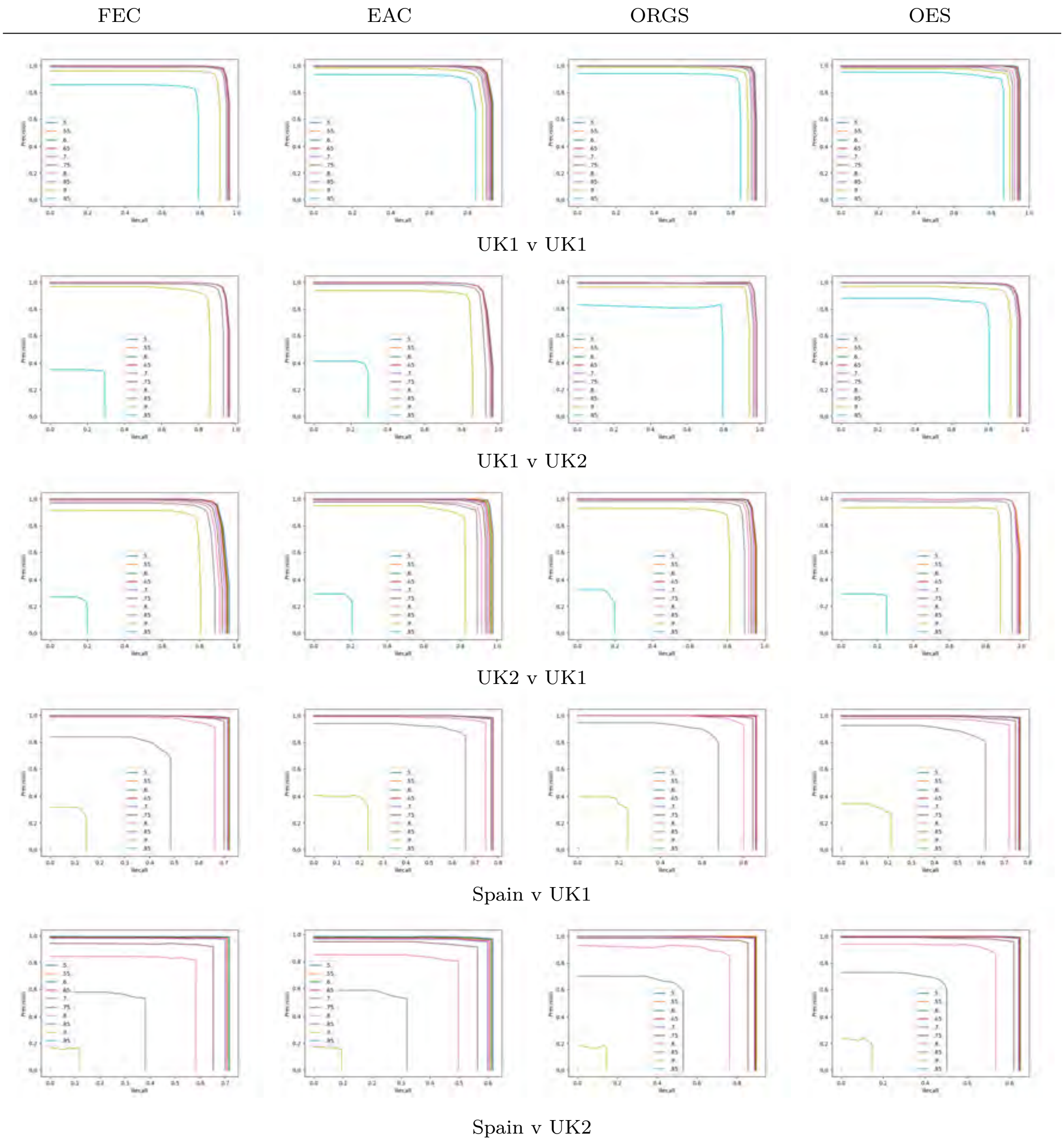
The PRC can be summarised using the Average Precision Score (APS), defined as the weighted mean of the Precision computed at each confidence threshold, by using the Recall from the previous threshold as the weight: $AP = \sum_n (R_n - R_{n-1}) \cdot P_n$, where R_n is the Recall and P_n is the Precision computed at the n-th classifier confidence threshold.

The PRC represents a valuable tool for performance analysis given the intrinsic class distribution in the datasets used in the experiments presented in this Chapter. It is also useful for comparing the detection performance of broccoli detection pipeline to the results published in the literature.

The SVM classifier performance was measured via different confidence and IoU thresholds. For each IoU threshold, we calculate the PRC of all predicted results. Figure 3.12 shows a group of selected examples of the APS computed at different confidence threshold settings of the broccoli heads detection system with the OES, ORG, EAC and FEC clustering algorithms for different IoU threshold values.

Every PRC plot in Figure 3.12 shows a higher area under the curve for smaller IoU threshold values. This is expected, but with a single threshold value for the IoU metric, Precision and Recall may not express how much of the broccoli heads was precisely located by the different algorithms. To analyse this further, the mean Average Precision (mAP) is used, calculated by averaging the APS over multiple IoU values within the range 0.5-0.95 in 0.05 steps [Lin et al., 2014].

The mAP yields a value close to zero when the extracted 3D points are less precisely located on a broccoli head, and a value close to one when the points are more precisely located. This evaluation complements the traditional APS computed at a



FEC EAC ORGS OES

UK1 v UK1

UK1 v UK2

UK2 v UK1

Spain v UK1

Spain v UK2

Figure 3.12: Selected PRC plots showing the classification performance of the clustering algorithms on different training and testing datasets combinations. Each curve is the PRC at various discrimination threshold settings for different IoU values within the range 0.5-0.95.

single IoU of 0.5. A complete set of APS values for various datasets combinations between the four clustering methods is summarised in Table 3.3.

	FEC			EAC			ORGS			OES		
	UK1	UK2	Spain	UK1	UK2	Spain	UK1	UK2	Spain	UK1	UK2	Spain
UK1	98.2	96.3	95.5	98.1	97.0	95.1	98.4	96.7	95.4	99.3	97.6	94.9
UK2	97.7	99.1	96.6	96.2	98.1	96.0	95.6	98.1	95.3	97.4	99.0	95.3
Spain	90.2	89.7	95.9	93.5	93.2	97.2	93.9	94.1	98.0	93.9	94.2	97.2
μ	95.4	95.0	96.0	95.9	96.1	96.1	96.0	96.3	96.2	96.9	96.9	95.8
$\overline{\mu} : \sigma_{\overline{\mu}}$	95.5 : 3.1			96.0 : 1.7			96.2 : 1.6			96.5 : 1.9		
UK1	96.1	92.2	94.3	96.9	94.3	95.1	95.8	95.0	95.7	97.1	95.1	96.1
UK2	92.4	92.1	93.1	93.1	92.6	93.5	93.3	92.9	93.4	94.5	93.1	93.9
Spain	56.3	61.3	65.2	68.2	73.1	74.6	67.3	74.7	75.1	71.4	75.2	76.3
μ	81.6	81.9	84.2	86.1	86.7	87.7	85.5	87.5	88.1	87.7	87.8	88.8
$\overline{\mu} : \sigma_{\overline{\mu}}$	82.6 : 15.5			86.8 : 10.7			87.0 : 10.6			88.1 : 9.9		

Table 3.3: Classification performance on various dataset combinations. The top half of the table lists the Average Precision Score (APS) values for an IoU of 0.5, while the bottom half shows the mAP averaged across all IoU values in the range [0.5...0.95]. μ represents the mean values, $\overline{\mu}$ is the mean of μ, and $\sigma_{\overline{\mu}}$ is the standard deviation of $\overline{\mu}$.

Similar to the F1 Score evaluation, the overall mAP is consistent for the UK set throughout the evaluation, but it decreases for the Spain dataset. This is caused by the difference between the area under the PRC for the lowest and highest IoU values.

The mAP is higher for smaller IoU threshold values, but the difference between the PRC plots shows that the overlap is more significant for the curves that are depicted closer together in Figure 3.12. However, the non-segmented areas mainly affect smaller regions and the overlapped area is sufficient for harvesting crops of marketable size at overlaps of 0.5 or higher. Moreover, a robotic harvester offers multiple opportunities for detection, as small-sized heads will grow to the desired size and will be harvested in another field pass.

Also, the experimental results on the two datasets show how the variability of the distance between the sensor and the crop affects the detection success rate. This is more evident in instances where the Spain set was used for training and the UK set for testing. These results may suggest some guidelines for the hardware configuration of an autonomous robotic broccoli harvester.

Table 3.3 shows a reduction in precision as the IoU threshold increases, especially for the FEC and the EAC algorithms, both based on similar distance strategies. These results, however, can be favourably compared with the APS of 95.2% and 84.5% for the same UK and Spain datasets reported by [Kusumam et al., 2017], and the AP of 83.5% reported by [García-Manso et al., 2021]. They can also be compared with the mAP for an IoU of 0.5 of 94.3%, and an mAP for an IoU of 0.75 of 89.51% published by [Bender et al., 2020], as well as the mAP for an IoU of 0.5 of 91.0%, and for an IoU of 0.7 of 83.0% recently published by [Psiroukis et al., 2022].

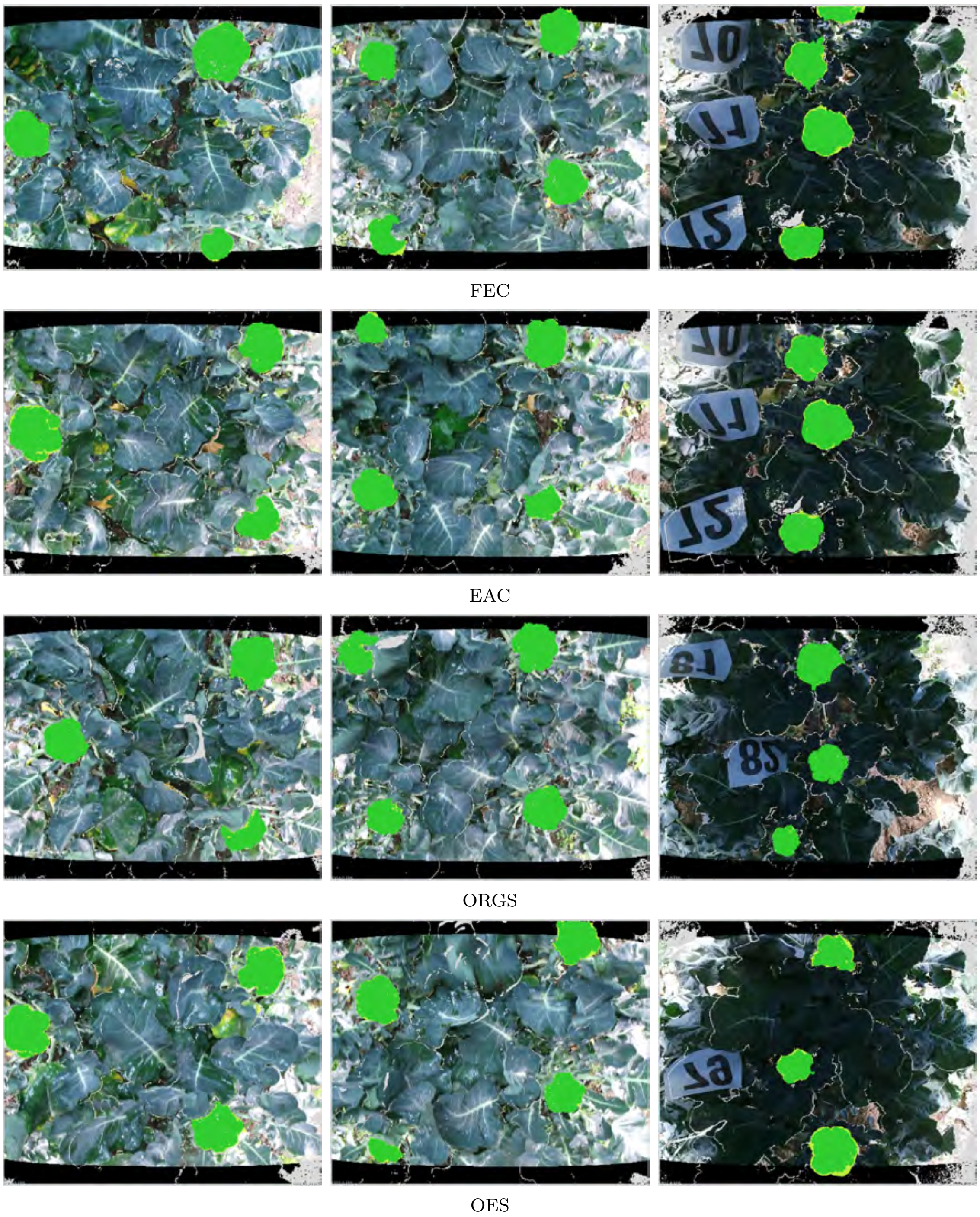
Figures 3.13 and 3.14 shows a set of selected examples for all four clustering algorithms of some of the best and faulty detection results, respectively. A large number of the false negatives shown in Figure 3.14 were on broccoli heads only partially visible in the frame, mostly happening on the edges. Nevertheless, these detection results were also part of the overall performance evaluation of the experiments presented in this chapter and calculated for the total number of broccoli heads.

3.2.2.3 Time Performance

One relevant feature of the FEC, EAC, ORGS and OES algorithms is that they achieve processing frame rates of 9.39, 10.99, 10.38 and 14.56 fps, respectively, running on an Intel i7 processor at 3.7 GHz clock speed. This processing time is a significant improvement on related research [Blok et al., 2016, Kusumam et al., 2017, Blok et al., 2020, Blok et al., 2021] and is the result of how the 3D space is explored by the algorithms.

While FEC and EAC search for neighbouring points in Euclidean space by first projecting the points on the organised structure of the point cloud into the 3D space, which takes some extra time, both the OES and the ORGS methods can retrieve any point information in constant time by directly exploiting the organised grid layout of the point cloud.

Because other sub-systems in the robotic harvester (e.g., grasping, cutting, navigation, etc.) also require time to perform their own operations, an efficient detection

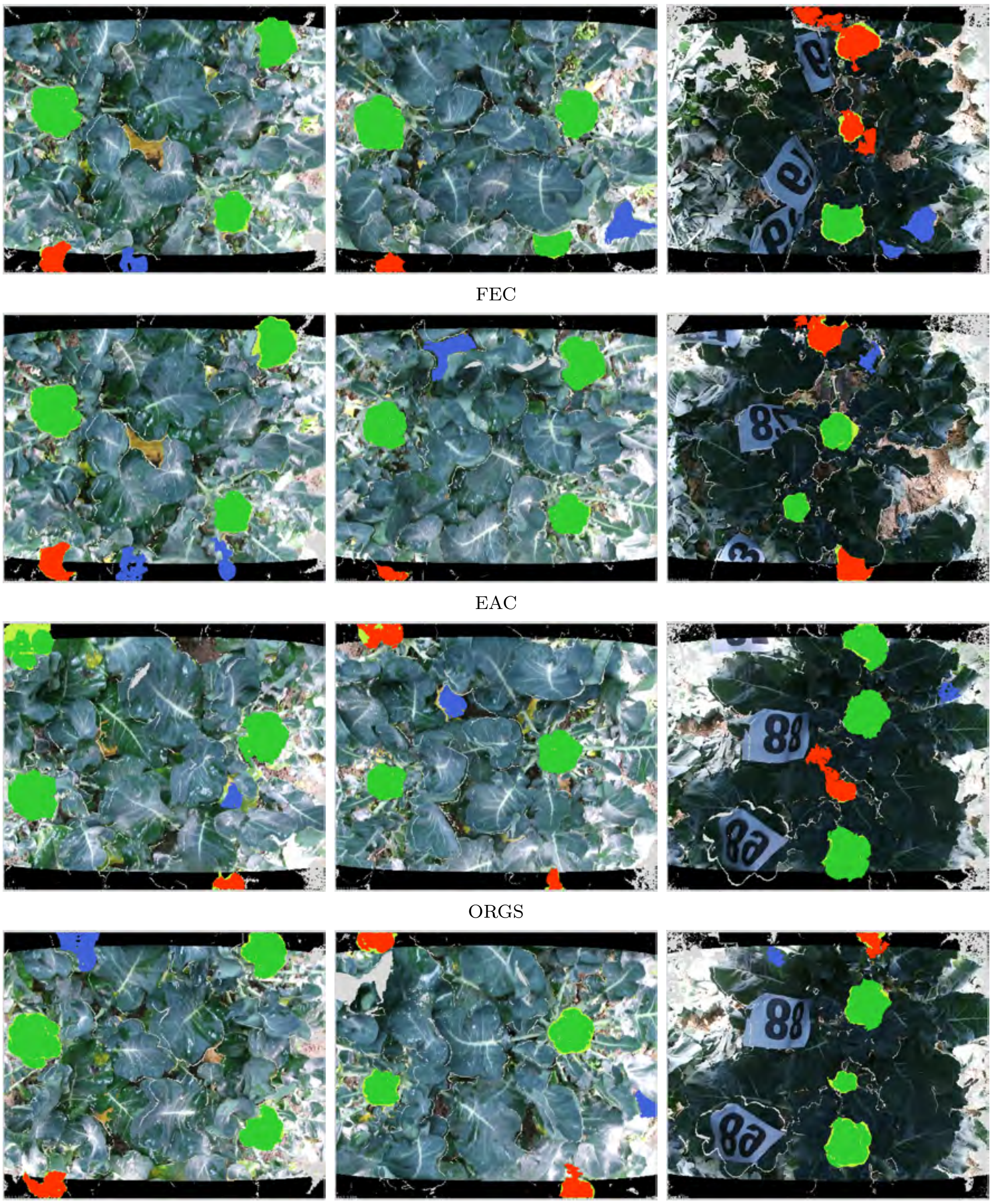

FEC

EAC

ORGS

OES

Figure 3.13: Selected examples of some positive detection results of the four algorithms on both datasets. The green areas show the true positive detected locations according to ground truth annotations (shown in yellow underneath the green layer).



FEC

EAC

ORGS

OES

Figure 3.14: Selected examples of some mixed detection results of the four algorithms for both datasets. The red areas, either over or under segmented, show the false negative detected locations, while the green areas show the true positive detections, and the blue areas show the false positive locations.

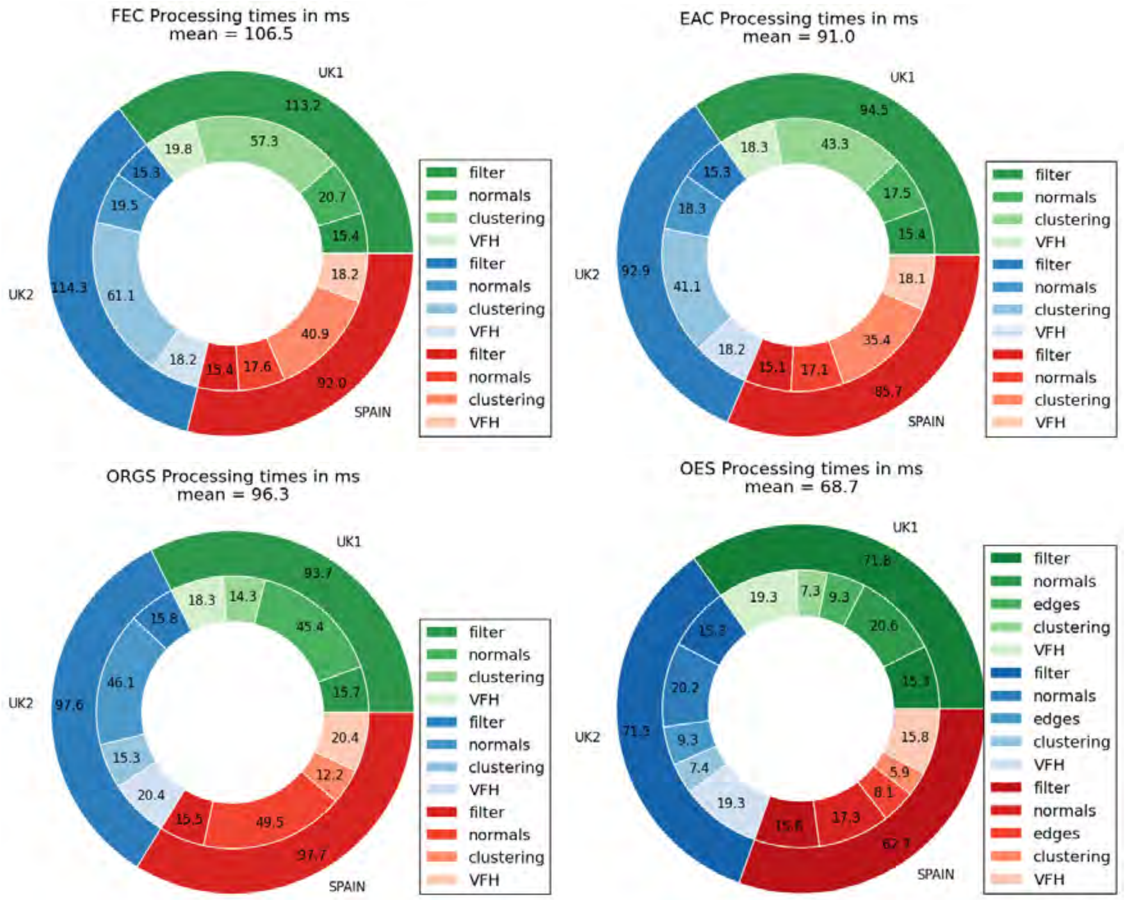


Figure 3.15: Average execution times of the broccoli detection pipeline when using the different clustering methods. The inside ring of each plot shows (in shades of the same colour) the processing time in miliseconds (ms) taken by each stage of the pipeline, whereas the outside rings show the total time taken by the entire process for each of the three datasets. The mean time for all datasets is shown at the top of each plot.

system benefits its overall performance. Thus real-time operation is one of the crucial requirements of autonomous robotic harvesting applications to increase yield and reduce other costs. Figure 3.15 shows the average execution time of the pipeline when using the four broccoli head clustering methods.

3.3 Conclusions

The classification processing speed and the training time are two factors that make 3D features and conventional machine learning methods stand out in the era of deep learning. The classification results show that the sensor distance used in the UK and Spain datasets produces a high broccoli head detection rate and suggests an appropriate hardware setup for the robotic selective harvester. The results show that the Spain dataset is more difficult than the UK one. This is mainly due to the high number of occlusions of the broccoli heads. However, cross-validation of both datasets indicate a high generalisation performance of the pipeline under different field conditions for the two broccoli varieties.

Comparative experimental results also show that the methods presented in this chapter achieved both high classification performance and real-time execution against recent approaches for broccoli detection available in the literature, either based on the Euclidean proximity of 3D points when tested on the same datasets, or based on conventional machine vision methods. These results are also comparable with the most recent deep learning models, even though the methods detailed in this chapter still present some generalisation shortcomings. The clustering strategies implemented by all clustering algorithms yield a trade-off between area segmented and detection accuracy, as the size of the clusters extracted provides enough information for harvesting the most marketable heads.

The evaluation performance shows that the algorithms exhibit the required detection accuracy and real-time performance needed for autonomous robotic harvesting applications. Nevertheless, other improvements can be adopted to further enhance the generalisation of the clustering algorithms in particular, and of the detection pipeline in general. An interesting approach would be to adopt strategies to better encode the properties of the broccoli heads to achieve a more accurate clustering of 3D points. This is important to estimate more precisely the size of broccoli heads suitable for today's market standards. However, this might also constitute a limitation, as machine learning algorithms based on pre-designed features involved skill and effort whose generalisation capability are either limited or enhanced by the SVM classifier used in the pipeline and its various configuration settings.

SVMs are one of the most robust prediction methods largely characterised by the choice of the kernel [Sanchez, 2003], which is perhaps their biggest limitation. Once the kernel has been chosen, SVM classifiers have only the error penalty parameter to adjust, but the kernel may involve many other parameters to fiddle with that are often left alone [Bartlett et al., 2002]. In this sense, the choice of the best SVM kernel for a given problem and the delicate and computationally expensive hyper-parameter tuning are still an open research problem [Meyer et al., 2003, Potts and Schmischke, 2022].

Additionally, data augmentation and regularisation techniques improve the overall performance evaluation [Polson and Scott, 2011]. However, this may lead to an added limitation of speed and size of training samples, as training for very large set of descriptors is also a further area for research due to the large number of support vectors produced for a trained model.

Nevertheless, based on the above line of reasoning, this dissertation has also investigate Convolutional Neural Networks and related deep learning techniques for broccoli head segmentation, as they have become the method of choice for many detection and classification problems. This will be addressed in the following chapter.

4 Deep Learning-based Detection and Segmentation of Broccoli Heads

4.1 Background and Motivation

The surge and interest in deep learning methods for perception has greatly improved performance in a wide variety of tasks in agriculture and many other fields. Automation in agriculture, however, presents very different and often more challenging scenarios than indoor and industrial environments due to the cluttered and constantly changing conditions of field farms.

A limitation of the machine learning pipeline presented in Chapter 3 is that the clustering algorithms are based on a predefined set of features that showed a generalisation performance heterogeneity for the different broccoli varieties of the datasets. This feature engineering process uses domain knowledge and design effort to extract descriptive properties from the 3D data, rather than supplying only the raw data to the machine learning process. The process followed for manual feature engineering depend entirely of the problem at hand and it is usually revisited for each new problem. Automated feature engineering improves upon this common workflow by automatically extracting useful features from a dataset, and is more efficient and repeatable as it tends to build better predictive models. Automation of feature engineering is a process inherent to deep learning algorithms [LeCun et al., 2015].

The wealth of research studies available in the literature shows that deep learning algorithms currently provide state-of-the-art performance for several crop detection tasks [Oliveira et al., 2021, Silwal et al., 2021, Yang and Xu, 2021]. Similarly, [Kamilaris and Prenafeta-Boldú, 2018] have shown that deep learning algorithms outperformed pre-designed feature-based algorithms in the 22 agricultural cases studied.

In line with these results, this chapter presents a deep learning approach using 3D information for detection and segmentation of broccoli heads based on Convolutional Neural Networks (CNNs). In the manner of Chapter 3.2.2.2, the method also exploits the organised structure of the point clouds captured by affordable RGB-D sensors. The method achieves comparable performance than recent published results, with high accuracy and generalisation in unseen scenarios, whilst significantly reducing inference time, making it better suited for real-time in-field applications.

Conventional 3D vision systems have been successfully applied to numerous applications featuring relatively large objects with distinguishable shapes such as furniture, rooms or offices [Qi et al., 2017, Li et al., 2018, Jiang et al., 2018]. When applied to an agricultural context in outdoor scenarios, however, these methods struggle to achieve satisfactory results, particularly for small objects [Le Louedec et al., 2020], due to the noisy and sparse distribution of the 3D data collected with low cost RGB-D sensors. In this study, we propose to overcome this limitation by exploiting the organised structure of the RGB-D point clouds, and implement a CNN-based architecture to learn detection and segmentation of broccoli heads. The CNN learns shape and diverse information from the point cloud by using the organised point cloud matrix layout as a medium for grouping and sampling 3D points.

The CNN is directly applied on the 3D points and their surface normals to extract localisation, shape, and broccoli head structures. This approach addresses the problem encountered with unorganised approaches [Qi et al., 2017], with a faster inference time and better feature extraction. The method achieved a high performance in terms of accuracy, segmentation, and localisation, with a better generalisation for the most difficult datasets at processing speeds of 50$\sim$60 fps.

The results of this chapter have been published in [Louedec et al., 2020].

4.2 Deep Learning

Deep learning models can be referred to as neural networks with deep architectures [Zhao et al., 2019]. Neural Networks (NN) were originally conceived to simulate the neurone system of the human brain to solve general learning problems. Its popularity peaked in the 1980-1990s with the appearance of the *back-propagation* training algorithm [Rumelhart, 1986], but due to the overfitting issues, lack of training data, limited computation power, and lower performance compared with other machine learning techniques, NNs became stagnant in the early 2000s. NNs and deep learning models rose up again due to the emergence of large-scale annotated training data, and also due to the development of high-performance parallel computing systems, as well as advances in the design of network structures and training strategies, such as pre-trained auto-encoders and data regularisation and augmentation techniques to reduce overfitting issues [Joshi and Mewada, 2020].

Convolutional Neural Network (CNN) with different degrees of modifications is the most representative model of deep learning [LeCun et al., 2015]. Traditional machine vision has struggled to reach the performance of a two year old child, but modern Convolutional Neural Networks have surpass human-level classification performance on restricted domains [Szeliski, 2022].

A typical CNN architecture consists of several layers: an input layer, a number of hidden layers and an output layer. Each layer in the network can be viewed as a specific feature map. Typically the input layer of the CNN performs a dot product of the convolution kernel with the layer's input matrix (with shape: number-of-inputs · input-height · input-width · input channels). As the convolution kernel operates on the input matrix, a feature map is generated, which in turn contributes to the input of the next layer. The middle layers are called hidden because their inputs and outputs are masked by convolution operations and activation functions (*i.e.*, functions that transforms the weighted sum of the inputs of a node or *neurone* into an output value to be transferred to the next layer nodes or as the network's output). This is followed by other layers such as pooling layers, fully connected layers and additional convolutional layers. Pooling layers, summarise the responses of a previous layer into one value to produce more robust feature descriptions, while fully connected layers connect every node in one layer to every node in another layer. With an interleave between convolution and pooling, an initial feature hierarchy is constructed that can be fine-tuned by adding several fully connected layers to adapt to the task at hand. A final layer with different activation functions is added to get a specific conditional probability for each output node. The feature extraction of CNNs are internally optimised during training on an objective function via the stochastic gradient descent method.

Recent advances in the field of deep learning greatly improved detection and segmentation tasks, making it more robust to the challenges involved. In this sense, deep learning detection algorithms have been shown to be robust to crop appearance and outdoor conditions variability, as they generalise well to new crop varieties and field environments [Kootstra et al., 2021, Silwal et al., 2021].

4.3 Method

The approach taken in this chapter for the segmentation and detection of broccoli heads uses organised point clouds originating from RGB-D sensors. A CNN auto-encoder is trained for the task of semantic segmentation using 3D information. To avoid over-fitting and to improve generalisation between different broccoli varieties, several data augmentation techniques are used. The segmentation results are transformed into instances using the connected components algorithm. Figure 4.1 provides a general overview of the system and indicates the core components described in detail in the following sections.

4.3.1 Pre-processing Organised Point Clouds

The most popular approaches for 3D object segmentation and detection were designed for processing unordered point clouds, such as in PointNet++ [Qi et al., 2017] and PointCNN [Li et al., 2018]. These methods rely on sampling and grouping points to learn surfaces, manifolds and shape. Such approaches, however, struggle with the noisy, cluttered and complex 3D information commonly found in agricultural applications [Le Louedec et al., 2020]. An alternative is to adopt the organised structure of the point cloud data captured by modern low-cost RGB-D sensors. This can be achieved either by segmenting directly in 2D and projecting the values into 3D space as in [Zeng et al., 2017, Blok et al., 2021], or by directly processing 3D information in the grid of the matrix layout.

A similar approach has already been explored in a previous work published by [Li, 2017]. In that study, authors chose to represent the point cloud data in a grid and to encode occupancy with a simple Boolean value. Based on this grid and using a standard CNN architecture, the experimental evaluation achieved significant

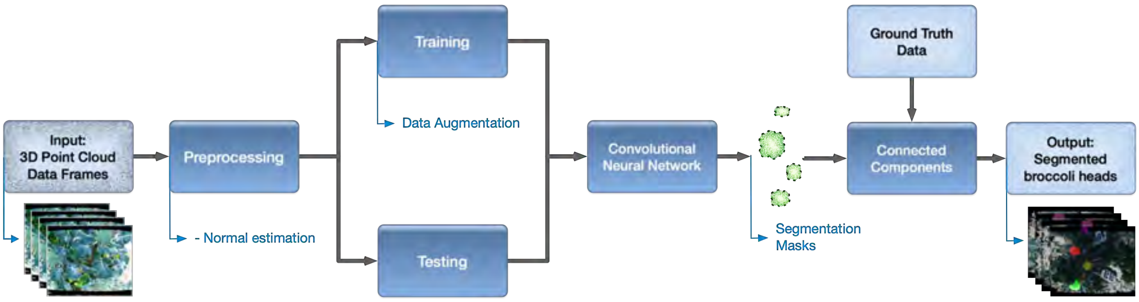


Figure 4.1: An overview of the CNN-based broccoli heads detection and segmentation pipeline for organised point clouds. The input data are the point cloud frames acquired by the RGB-D sensor. The output is the segmented broccoli heads on each input frame.

performance improvement over previous point cloud based detection approaches using the KITTI dataset, a standard benchmark for 3D vision in autonomous driving [Geiger et al., 2012].

Typical methods for unordered point clouds make considerable use of 2D convolutions to learn local features from sampled and aggregated points [Blok et al., 2021, Kootstra et al., 2021, Silwal et al., 2021]. The spatial information and correlation is highly dependent on the sampling and grouping algorithms, which is an active area of research [Jiang et al., 2018, Yang and Xu, 2021, Zhang et al., 2021]. For RGB-D sensors, however, the data is captured from a single point of view and can readily produce point clouds with spatial organisation and grouping of points with an organised grid structure, as detailed in Section 3.1.1.3 of Chapter 3. Based on these preliminary assumptions, 2D convolutions and conventional CNN architectures can directly be applied to a 3D point cloud through its organised grid layout.

In this chapter, we hypothesised that clusters of broccoli head clusters can be retrieved successfully from these organised point clouds by using surface normal features, convolutions and pooling functions. Processing 3D points using convolutions should lead to filters dedicated to both computing these surface normals and extracting relevant information from them. This information along with spatial data should improve the learning of shape and other local features by the neural network. In line with this hypothesis, only the 3D point data and surface normals are used as input for the CNN broccoli segmentation pipeline. In this sense, surface normals are computed based on the same Integral Images Normal Estimation method described in Section 3.1.2.1 from Chapter 3. Values missing from the input point clouds, often represented as NaN, are promptly replaced by the $[0.0, 0.0, 0.0]$ vector on both the 3D points and their corresponding surface normals. The time overhead for normal estimation was of 44 ms per frame on average.

All experiments reported in this chapter have been implemented using the algorithms available as part of the Open3D library [Zhou et al., 2018] for processing point clouds.

4.3.2 Convolutional Neural Network

For the semantic segmentation task, a classic auto-encoder architecture has been chosen inspired by U-net [Ronneberger et al., 2015]. U-Net is a CNN architecture developed for semantic segmentation of biomedical images based on a modified architecture of the Fully Convolutional Network to work with fewer training samples and to improve segmentation results [Shelhamer et al., 2017].

U-Net consists of an encoding part, for extracting relevant features, and a decoding part, for transforming the extracted features into the correct class prediction. The encoder follows the typical architecture of a CNN, which repeatedly applies two 3x3 unpadded convolutions, each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling. At each downsampling step, the number of feature channels is doubled and each step of the decoding part consists of (a) an upsampling step of the feature map followed by a 2x2 up-convolution that halves the number of feature channels, (b) a concatenation with the corresponding cropped feature map from the contracting path, and (c) two 3x3 convolutions, each followed by a ReLU.

The cropping is added due to the loss of border pixels after applying each convolution. At the final layer, an 1x1 convolution is used to map each of the 64-component feature vector to the desired number of classes. In total, the architecture of the network has 23 convolutional layers. The idea behind U-Net is to extend the encoder by successive layers, where pooling operations are replaced by upsampling operators. A successive convolutional layer can then learn to assemble a more precise output based on this information.

As depicted in Figure 4.2, connections are added between the encoding and decoding part of the architecture to use multi-scale features in the segmentation process. The input consists of 6 features, *i.e.*, (x, y, z) location of each point and its corresponding

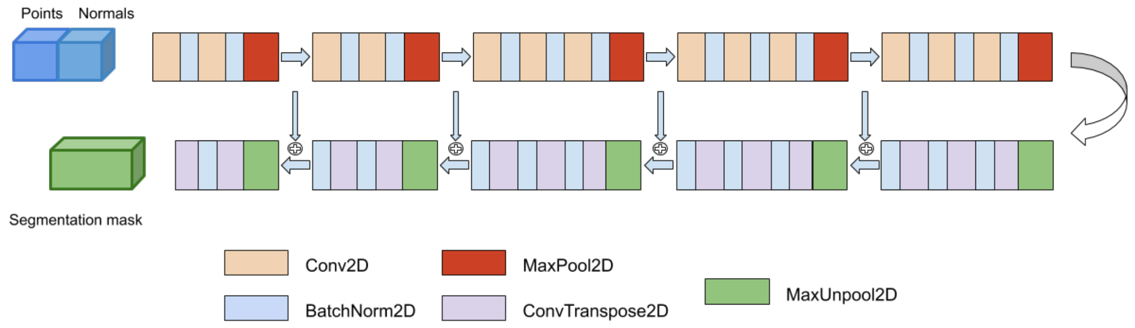


Figure 4.2: The CNN architecture employed for the segmentation of organised point clouds.

3-component surface normal vector, which are compressed into a $512 * W * H$ feature map in the latent space of the network, before being decoded into the segmentation mask.

The standard VGG16 architecture [Simonyan and Zisserman, 2015] is used for the encoding part and the feature maps and pooling indices are saved at the end of four different convolution blocks. These indices are then used in the decoder part of the network to upsample the feature maps and to introduce multi-scale features and improve the extraction of broccoli heads clusters. As a result, the inference time of the relatively compact nature of the CNN is above 50 frames per second.

Because the CNN architecture used in this study was developed for semantic segmentation tasks, a connected component algorithm was used over the points in the point cloud grid, processed as a binary mask. Clusters considered too small to be broccoli heads (less than 200 points) were removed, and segmented values with an IoU < 0.5 (see Section 3.2.2.1) were also discarded. A similar procedure was adopted in a point cloud segmentation study preliminary to this dissertation [Montes et al., 2019].

4.3.3 Data Augmentation

Training the network only on point coordinates leads to fast learning with a quick convergence toward dataset specific over-fitting. Training only on one dataset leads to excellent results (~ 0.99 mean Average Precision) but offers little to no generalisation when applied to unseen scenarios. As the position of broccoli heads in each dataset is changing linearly, the CNN tends to learn their position and change in location. To prevent this, the normal information is first added on top of the points coordinates, and then various data augmentation techniques are applied during the training phase.

Since the point cloud data is contained on a 2D matrix layout, rotations in this grid can be readily applied to avoid localisation over-fitting. Similar rotations also need to be applied to the data points and surface normals to avoid discrepancies between the grid and the spatial coordinates and orientation. Translations over the points can also be included, adding more diversity to the object positions. These three augmentations allows the CNN to add more diversity to the broccoli heads locations. For every frame in the training set, the points are rotated in space and in the grid by a random angle between -180 and 180 degrees. In addition, the point clouds are also translated by a random value between its minimum and maximum on every axis.

4.4 Training and Evaluation Setup

Without proper and large datasets, machine learning algorithms cannot succeed. By modelling crops of broccoli plants, a higher level domain knowledge can be added to the entire learning process, potentially increasing the detection and segmentation performance. The CNN training dataset consisted of randomly selected point cloud frames from each of the broccoli datasets described in Section 3.1.1.1 of Chapter 3. Each of the datasets is split into two different sets for training and testing. A proportion of 75% frames with the annotated data were used for training (from which a validation set of 25% was extracted), and the remaining 25% were used for testing. The training set was used to adjust the network weights and the validation set was used during training to control the learning rate of the network to reduce overfitting. This training-testing scheme allows the generalisation performance analysis of the algorithm and its shortcomings.

During training, the Adaptive Moment Estimation (Adam) optimiser [Kingma and Ba, 2014] was used. Adam is an adaptive learning rate optimisation algorithm designed for training deep neural networks. As such, it computes individual learning rates for different parameters of a deep learning model. Adam can be used instead of the classical stochastic gradient descent procedure to iteratively update network weights based on training data. For the experiments conducted for this chapter, the learning rate was started at 0.0001 and then reduced by 0.7 every 200 epochs. Training is eventually stopped when the loss rate had significantly decreased. The hardware used for training and testing the neural network was an NVIDIA 1080Ti GPU and an Intel i7 4790 for the CPU code, which handles I/O operations and the pre-processing stage of the pipeline.

For the application of perception algorithms to harvesting, the main interest is in the accuracy of the deep learning algorithm, its precision, its generalisation performance between different locations and broccoli varieties, and its inference speed. In this sense, the performance of the method was evaluated for both *semantic* segmentation and *instance* segmentation. Semantic segmentation associates every point of a cloud with an annotated class label, either broccoli or background. Multiple objects of the same class are considered to be a single entity. In contrast, instance segmentation handles multiple objects of the same class as distinct individual instances. Aiming for a higher degree of accuracy, the broccoli datasets used in this thesis were all annotated for instance segmentation as detailed in Section 3.1.1.2 of Chapter 3.

The underlying reasons for this evaluation are twofold: Firstly, the quality of the broccoli heads detection and their accuracy in terms of missing detection and false detection can be better reflected. Secondly, the mask from detection and extraction of the broccoli heads from the background is of fundamental importance for automated harvesting operations to properly determine relevant features such as size and shape and to better assess the quality of the masks extracted.

Two different metrics are used to evaluate the segmentation performance of the CNN: the mean Average Precision (mAP), as discussed in Section 3.2.2.2 of Chapter 3, and the mean Intersection over Union (mIoU), defined as the average of the IoU between the segmentation masks returned by the CNN and ground truth data for all samples.

		OES			CNN		
		UK1	UK2	Spain	UK1	UK2	Spain
Test	UK1	97.1	95.1	96.1	94.8	93.3	76.3
	UK2	94.5	93.1	93.9	91.3	93.1	76.4
	Spain	71.4	75.2	76.3	79.1	81.2	87.4
	mean	87.7	87.8	88.8	88.4	89.2	80.0

Table 4.1: Comparison of the mAP for the CNN's instance detection masks and the OES clustering pipeline for all the datasets combinations. The average performance for each training set is also shown at the bottom row of the table. The differences in performance are mainly due to the intrinsec properties of the different broccoli varieties used in these experiments.

4.5 Results

In this section the CNN is compared to the results presented in Chapter 3 taking into account point and instance segmentation, and the performance of the solutions on the collected data is also assessed.

4.5.1 Instance segmentation

The standard metric for instance segmentation is the mean Average Precision (mAP). As stated in Chapter 3, it requires the assignment of a confidence score to each segment for computing a Precision-Recall curve (PRC) and the corresponding Average Precision Score (APS). The mAP is then calculated by averaging the APS over multiple IoU values within the range 0.5-0.95 in 0.05 steps. Table 4.1 summarises the different results achieved on the three broccoli datasets using both the CNN model in the segmentation pipeline and the OES algorithm (the highest scoring clustering algorithm presented in Section 3.2.2.2 of Chapter 3).

Overall both algorithms show a high performance for all training and testing scenarios. While both methods achieve a good performances across all datasets (*i.e.*, mAP$>$ 0.80), the CNN still falls behind on the Spain dataset. The OES seems to achieve a better generalisation when the Spain dataset is used for training and then tested on the UK set. In contrast, the CNN shows better performance for predictions when the Spain dataset is used for testing.

Lower results from the neural network can be explained by the missing or the partially detected broccoli heads visible on the edges of the point cloud frames. With a lower IoU, they impact negatively on the Precision and Recall metrics, as those segments are classified as either false positives or false negatives. Also, when trained on the Spain dataset and tested on the UK set, the lower score obtained by the neural network can be attributed to the extraction of small areas on leaves similar in shape to the smaller Spain broccoli heads.

The OES clustering algorithm extracts more segments that are found in the detection predictions later on, as seen in Figure 4.3. However, they rarely have a probability higher than 75% of being classified as a positive prediction. The lower performances from OES on the Spain dataset but higher when tested on the UK sets, are due to the segment extraction strategies of the algorithm, which struggles more on smaller and more occluded broccoli heads, such as those present in the Spain dataset, but performs better on big and separated crops, as those presented in the UK dataset. An example of this case is illustrated in Figure 4.4.

Figure 4.3 shows selected examples of challenging cases for the instance segmentation task when training the machine learning algorithms on the UK dataset and testing them on the more challenging Spain dataset. In that case, the neural network fails to detect some of the broccoli heads of shape and heavy occlusions not seen in the UK dataset. On the other hand, OES detects more of these challenging broccoli heads, but is not as discriminative and also detects False Positives on surrounding and background leaves and other similar areas.

Also with a high Precision and Recall, the CNN tends to be very selective in the choices for detection. This results in masks for the detected instances with high probabilities, but it also removes small detections in the Spain dataset when trained on the UK set, as small or partial broccoli heads are seen less frequently in the UK

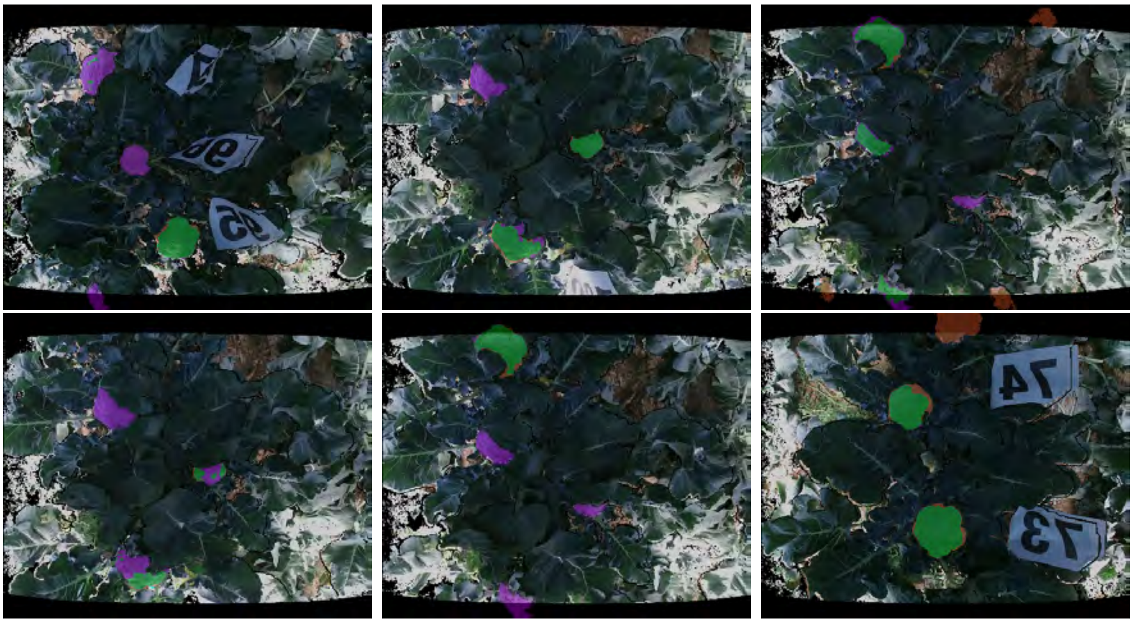


Figure 4.3: Challenging examples for both algorithms with the broccoli head prediction from the OES (top row) and the neural network (bottom row) pipelines. The colour overlay show the true positives in green, the false positives in orange and the false negatives in purple.

datasets. Figure 4.5 shows PRCs for both the CNN and OES pipelines when trained specifically on the Spain dataset and tested on that same set. The PRCs show that the CNN still exhibits a high performance for thresholds up to 85%, but achieves lower results after 95%, whereas OES starts to struggle earlier around 80%.

4.5.2 Semantic segmentation

Unlike instance segmentation, semantic segmentation does not require confidence scores to be associated with each extracted segment. For semantic segmentation, IoU is a commonly used metrics as it ignores object level annotations while considering only annotations at point level. Even though confidence scores and the IoU metric can be used together to evaluate the detection of an object, as shown in Section 3.2.2.1 of Chapter 3, IoU provides a finer per point measurement. Since broccoli instance annotations are not considered, the IoU metric alone cannot evaluate object classes. Conversely, the classifier's confidence scores and AP cannot measure the

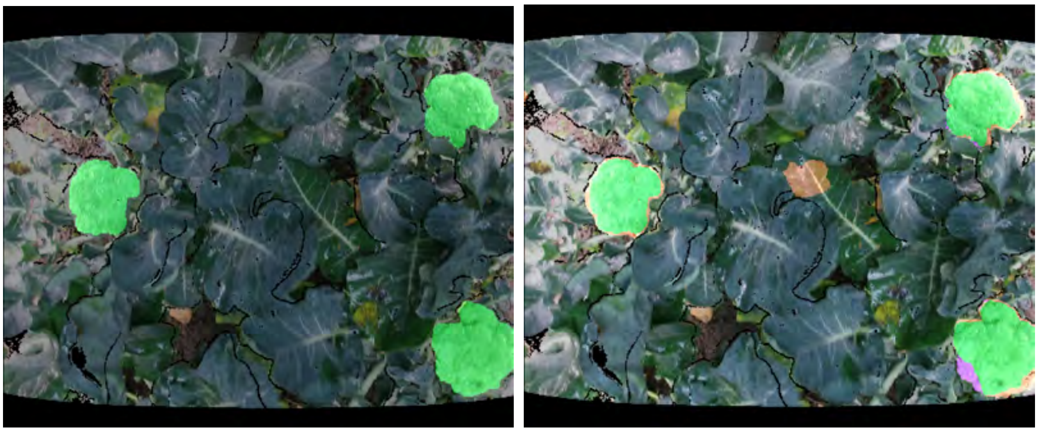


Figure 4.4: Prediction results when trained on the Spain set and tested on the UK set for the OES pipeline on the left, and for the CNN pipeline on the right. The performance of the OES is due to the clustering strategy of the algorithm which struggles on smaller and occluded objects, but performs better on the broccoli heads captured on the UK dataset.

		CNN			OES		
		UK1	UK2	Spain	UK1	UK2	Spain
Test	UK1	95.1	93.9	85.2	95.1	93.1	90.7
	UK2	92.4	94.2	85.4	92.1	92.9	93.1
	Spain	81.3	85.1	94.3	69.3	72.7	75.7
	mean	89.6	91.1	88.3	85.5	86.2	86.5

Table 4.2: Comparison of the mIoU for the CNN's segmentation masks and the OES clustering pipeline for all the datasets combinations. The average performance for each training set is also shown at the bottom row of the table.

output of semantic segmentation [Zhang et al., 2020].

The IoU is the area of overlap between the predicted segmentation and the ground truth divided by the area of union between the predicted broccoli head segmentation and the ground truth annotation. The IoU ranges from 0.0, meaning no overlap, to 1.0 meaning perfectly overlap segmentation. Table 4.2 summarises the results achieved by both the CNN and the clustering algorithms from Chapter 3 for the mIoU metric averaged over all segmented points in the tested datasets.

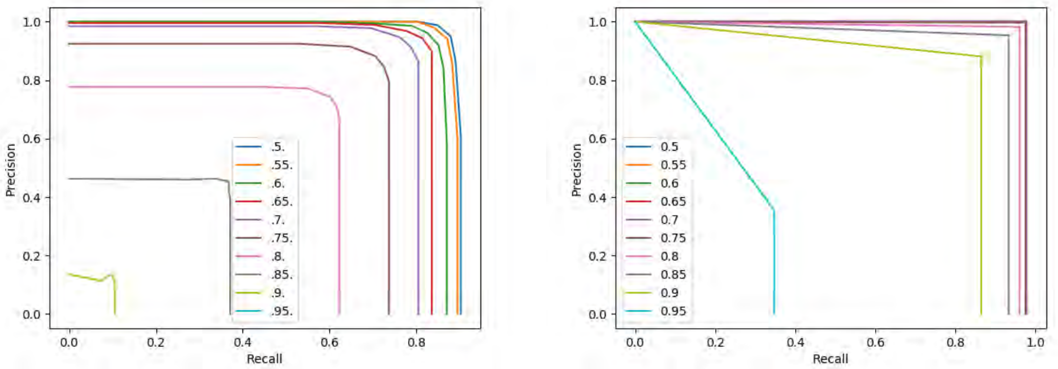


Figure 4.5: Precision-Recall curves for the OES on the left, and for the CNN on the right trained and tested on the Spain dataset with the IoU threshold in the range [0.5...0.95].

The neural network for this task shows improvements compared to the OES pipeline for all train-test combinations. This offers several advantages such as more accurate localisation, better shape estimation for tasks such as crop phenotyping and grasp prediction for harvesting the crops of broccoli plants. Furthermore, the results are consistent with Table 4.1, but some increments can be seen for the methods when tested on the Spain dataset. This is due to the false positive detection now being evaluated for their masks at point level, and impacted less than when they were evaluated for a whole broccoli head. That is, for a point cloud with four broccoli heads, ten false positives impact the score just around 400 points out of the 200k total points in the cloud.

Figure 4.6 and 4.3 depicts some prediction results from the CNN on the Spain dataset, which has presented the biggest detection and segmentation challenges for both pipelines. Both figures illustrate the differences in terms of mask and point classification, and how overall the convolutional neural network performs better on the UK dataset and not as well on the Spain set. In Tables 4.1 and 4.2, the CNN show similar differences in performances with those shown by the OES pipeline. While performing better or similarly well on most training-testing scenarios, the CNN struggles to generalise when trained on Spain and tested on UK. This is due to the broccoli heads size, as they produce more False Positives on the UK dataset where the scale differs and Spain-like broccoli shapes can be found on background elements such as leaves, foliage and soil.

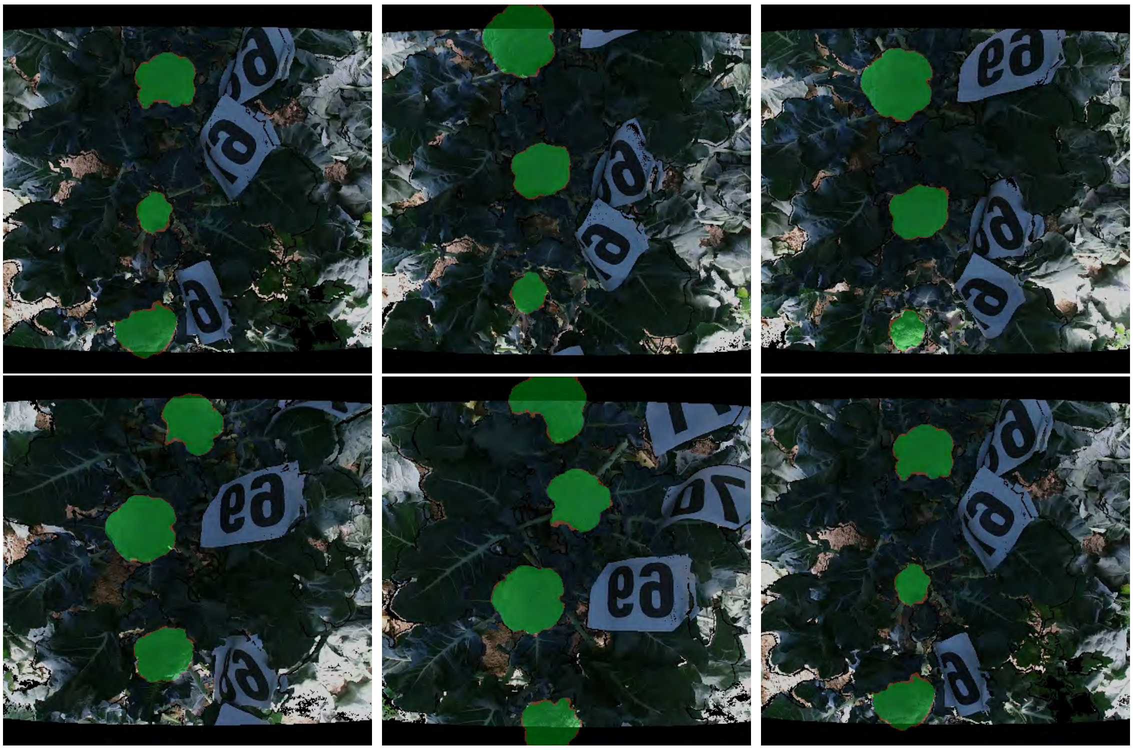


Figure 4.6: Example predictions of the CNN trained and tested on the Spain dataset. The colour layers correspond to true positive points shown in green, false positives shown in orange, and false negative points shown in purple.

4.5.3 Qualitative analysis

The qualitative analysis is performed on some examples of the Spain dataset, as they present a more challenging test than the UK dataset for both pipelines. Both algorithms were trained on the Spain dataset and tested on the same set. From Table 4.2, it can be seen that the neural network offers more reliable results at point level precision for the localisation and detection of the broccoli heads. Meanwhile, OES and the other three clustering algorithms struggle to extract small instances and tend to group together the broccoli heads with surrounding leaves, thus reducing their segmentation accuracy. Also, for bigger objects, the OES pipeline tends to generate more false negatives on the contours of the objects, while the CNN tends to add just a few more points to the mask.

Figure 4.7 shows different feature maps decoded by our network. Those feature

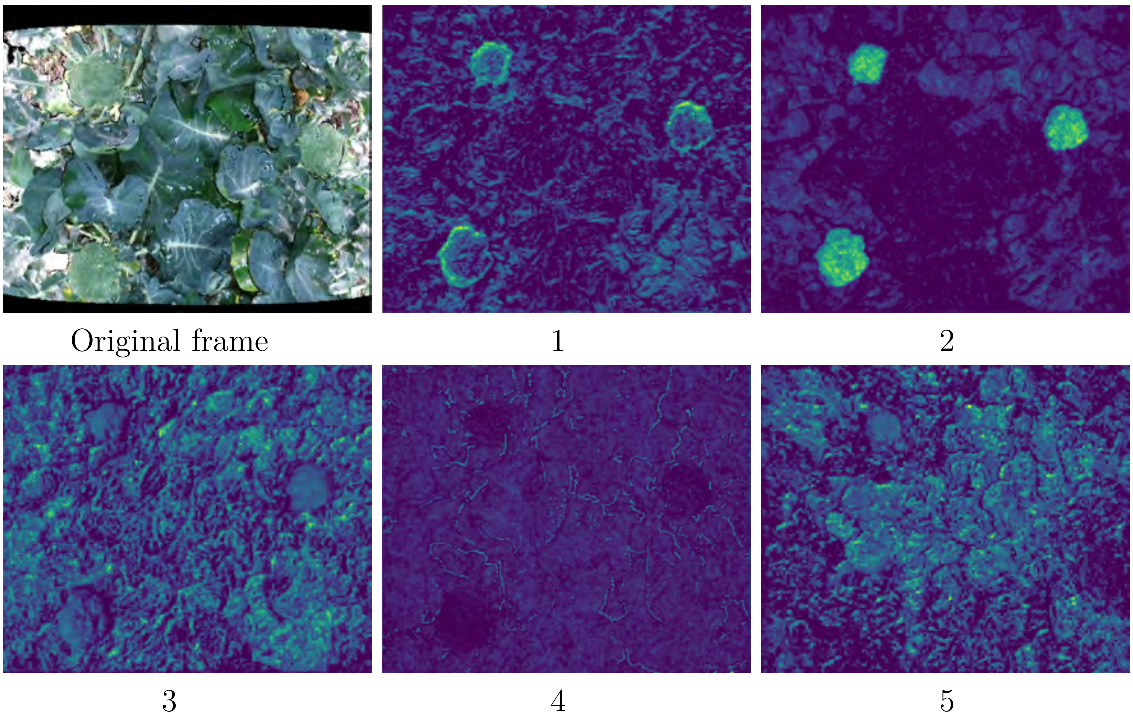


Figure 4.7: Examples of feature maps extracted by the CNN. The original point cloud is shown in the first frame. This frame is the input to the CNN along with the corresponding surface normals. The feature maps are extracted at the end of the decoder's de-convolution blocks.

maps are obtained by passing the point data and surface normals through the neural network and then visualising one of the dimensions of the feature map at the end of one of the decoder's de-convolution blocks.

The contour features characteristic of a broccoli head can be distinguished in the first feature map shown in Figure 4.7, which can be directly related to Figure 4.8, allowing a better extraction of the object by the network. In the second and third feature maps from Figure 4.7, features related to the shape of the broccoli heads are extracted, with emphasis on texture on the second map, and on the normal features on the third map. The fourth feature map shows the leaf edges extracted from the point cloud, while the last map shows features which seem to be more related to distances along the z-axis, with emphasis on vegetation above the ground.

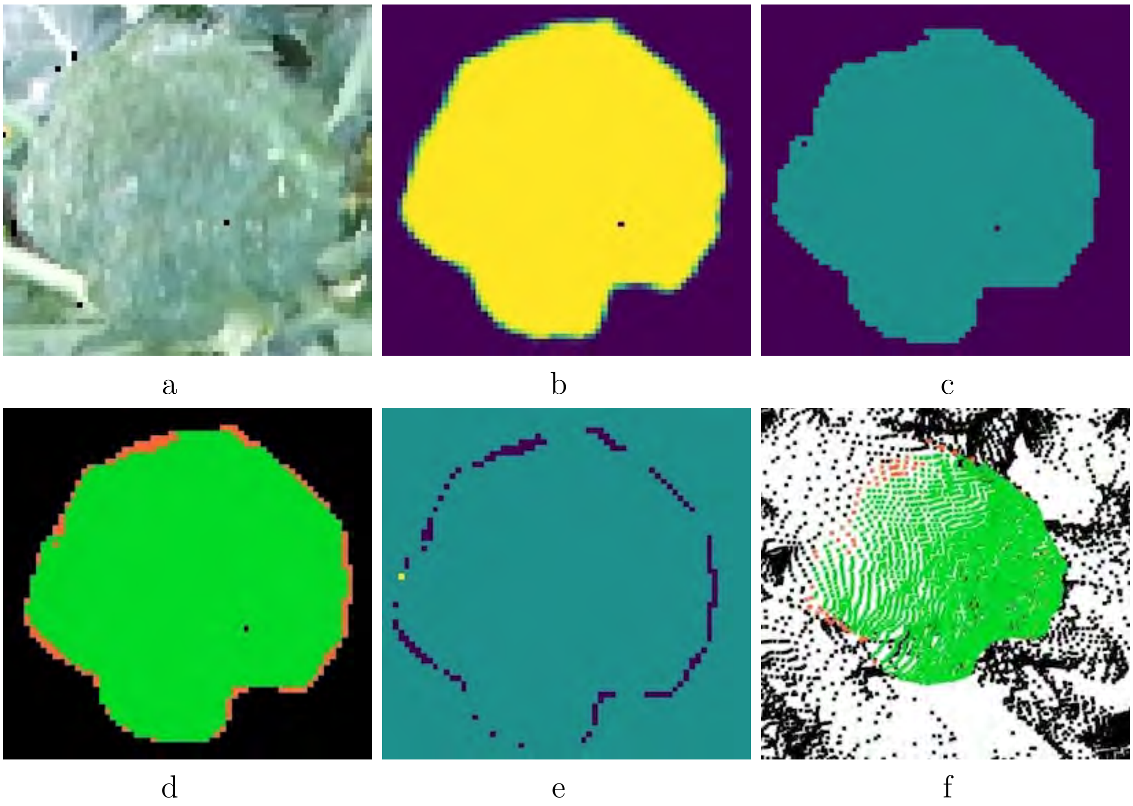


Figure 4.8: CNN's segmentation contour uncertainty. a) the original view of a broccoli head, b) the predicted segmentation mask, c) the ground truth annotation, d) the true positive (green layer) and false positive (orange layer) map, e) the difference with ground truth, f) the visualisation in 3D space of the final prediction.

Figure 4.8 depicts the uncertainty of the detection results for the CNN. Using the softmax function, the prediction uncertainty for each class (*i.e.*, broccoli and background) can be represented based on probabilities. The softmax function takes as input the CNN's output vector of k real numbers, and normalises it into a probability distribution consisting of k probabilities proportional to the exponentials of the numbers of the input vector. In this way, the larger numbers of the vector will correspond to larger probabilities. Segmented points with low probability were removed using a simple 0.5 threshold, some uncertain areas, however, still remain surrounding the segmented broccoli heads.

Figure 4.8(d) graphically depicts the lower probability surrounding the shape of a broccoli head sample. These low probability contours are most of the time either True Positive or False Positive, but can be readily discarded using a higher threshold.

However, even though not present in the ground truth annotation, those points provide useful information about the surrounding of the broccoli head and its width useful for further analysis. In addition, Figure 4.8(f) also shows the broccoli head's representation from a 3D perspective, where those contours can be used effectively to separate the broccoli head's points from other background points.

Finally, the total processing time of the CNN pipeline, including surface normals estimation and inference time, was of 200 ms on average per frame, equivalent to 5 fps, while the inference time alone of the CNN achieved processing speeds of 50∼60 fps. This difference comes from pre-processing the input data and not from the CNN predictions.

4.6 Conclusions

Based on the experimental results on instance and semantic segmentation of 3D information of broccoli heads acquired through RGB-D cameras presented in this chapter, some conclusions are straightforward to establish. Unlike the set of algorithms for real-time detection of broccoli heads presented in Chapter 3, deep neural networks do not require the design of handcrafted features, as deep learning algorithms have a powerful capacity of feature expression and also have an automatic feature extraction over large datasets. In addition, deep neural networks have the property of producing higher-level features by automatically composing lower-level features. This deep learning characteristic has been further extended in this chapter by directly providing additional low-level features in the form of surface normals on an organised arrangement of the input point clouds.

Deep learning algorithms are immediately appealing in terms of accuracy and overall performance, but they still present some challenges such as drifting (*i.e.*, the unforeseen changes over time of the target properties that the model is trying to predict, making predictions less accurate as time passes) and real-time processing. In particular, the CNN architecture presented here also faces some challenges intrinsic to the data. For instance, differences in size of the broccoli heads within the datasets lead to missed detections, especially on the top and bottom boundaries of the point cloud frames, where a large number of noisy data points appear and broccoli head size varies the most. Dealing with incomplete information due to occlusions is still a big challenge when operating in complex planted fields environments. Training on intrinsically different object sizes than the test set (*e.g.*, training on the Spain dataset and then testing on the UK set) also affect the results, yielding a higher number of False Positives.

These challenges can be addressed through some standard processes such as data normalisation, un-distortion, data augmentation, etc. Also, the use of more advanced architectures such as ResNet [He et al., 2016] could potentially improve robustness to scale and shape variation, occlusion and data diversity. The method achieves similar results than feature engineered clustering algorithms and better results for the most challenging datasets, while providing better segmentation of the broccoli heads, competitive instance segmentation, better localisation, and faster inference time. All these characteristics make the CNN pipeline well suited for real-time applications in autonomous selective harvesting, and open new possible applications in agriculture, as accurate crop detection is an essential component for tasks such as phenotyping, automated targeted spraying or automated crop monitoring. Also, crop detection is often the preliminary step to determine maturity level of crops for autonomous robotic harvesting operations.

In addition, because the same broccoli heads will be predicted multiple times, it is also a crucial component to associate detections between consecutive frames and uniquely track the same broccoli instance for other tasks such as crop counting and yield estimation. This is an essential component to supply accurate 3D localisation for the grasping and cutting systems of an autonomous robotics harvester. A tracking method that incorporates detection responses to handle these tasks is the subject of the next chapter.

5 Tracking Multiple Instances of Broccoli Heads

5.1 Background and Motivation

Crop detection and tracking are together an important part of autonomous selective harvesting as they can play a key role in the accuracy of the harvester's cutting system [Kootstra et al., 2020, Kootstra et al., 2021]. The high complexity found in planted open fields, tends to make harvesting success to drop, often because the cutting tool of an autonomous harvester is not able to reach or even to uniquely determine the correct location of the target crop [Mo et al., 2021]. In addition, throughout the years there has been a huge variability in the design of harvesting tools for autonomous selective harvesting in precision agriculture [Feng, 2021]. Consequently, detection and tracking, rather than being two separate tasks, should be part of an unified system towards improving generalisation and performance of selective harvesting systems.

This chapter describes a tracking method of multiple broccoli heads based on a *particle filter* [Elfring et al., 2021] that can be incorporated into an autonomous selective harvester. The approach combines a broccoli head detector and the particle filter to track multiple crops in a sequence of 3D data frames. All trajectories created by the tracker are verified based on a data association method that matches detections with tracks over each frame. This verification is carried out by introducing a registration step based on 3D feature vector data associations to avoid multiple tracking of the same crop observed in different frames. Furthermore, a track confirmation step is also introduced to deal with misdetections so that false negatives diminish while false positive trajectories decrease as well. Additionally, the particle filter incorporates a simple motion model to produce the posterior particle distribution, and a similarity model as probability function to measure the tracking accuracy.

A detection and tracking system should be able to generalise across variations in crop appearance while keeping high detection and tracking success rates. Moreover, to maintain a cost-benefit ratio acceptable by farmers and the whole agricultural industry, these tasks should be performed in real-time on modern commercial computing platforms. Our evaluation shows that the tracking method successfully reduces the number of false negatives produced by the detectors on their own. The method also accurately detects and tracks the 3D locations of broccoli heads relative to the vehicle at high frame rates.

5.1.1 Particle Filters for Tracking: Motivation

Since their introduction in in the early nineties [Gordon et al., 1993], Particle Filters have been among the most popular state estimation algorithms for the solution of iterative estimation problems. They do not require the assumptions of linearity and Guassianity like *Kalman Filter* based techniques and are capable of handling non-Gaussian noise distributions and non-linearities in the observable measurements as well as target dynamics. This is precisely the scenario of a broccoli planted field where locations of the crops need to be continuously and uniquely tracked for a successful harvesting operation. Additionally, The standard Particle Filter algorithm can be readily understood and used due to the widespread availability of tutorial material and software libraries [Speekenbrink, 2016]. Extensive research has advanced the standard particle filter algorithm to improve its performance and applicability [Shariati et al., 2019, Lan et al., 2020].

In comparison with standard approximation methods, such as the *Extended Kalman Filter*, a strong motivation for using Particle Filters is that they do not rely on any local linearisation technique or any functional approximation [Elfring et al., 2021]. Another reason is that for many large or high-dimensional problems, Particle Filters are tractable, whereas Kalman Filters are not. Often, however, the price to pay is computational, *i.e.*, a large number of samples are needed to closely approximate the estimated posterior states of the system. The more complex the models of the environment, the more state samples are needed to describe the posterior state distribution. Nevertheless, the availability of ever-increasing computational power, have seen particle filters being used in real-time applications in a diverse list of fields [Jinan and Raveendran, 2016, Wang et al., 2017].

In this chapter, a particle filtering technique is explored for multiple broccoli head tracking. This technique requires only a relatively small number of particles (*i.e.*, estimated broccoli location hypothesis) to help in reducing the added computational cost. An additional discussion on Particle Filter algorithms, including their advantages and some of their drawbacks, is provided in Section 5.5.

The results of this chapter have been published in [Montes and Cielniak, 2022].

5.2 Object Tracking

Object tracking is a process of estimating or predicting the positions and other relevant information of moving objects in a sequence of data frames once the initial position of the target object has been defined [Porikli and Yilmaz, 2012]. It has multiple practical applications ranging from daily life tasks to high level security uses, including education and augmented reality [Rambach et al., 2018], traffic monitoring [Lorenčík and Zolotová, 2018], robotics [You et al., 2019], autonomous vehicle tracking [Cao et al., 2021, Ravindran et al., 2021], surveillance [Joshi et al., 2018, Lo et al., 2021], among many others [Luo et al., 2021, Zhang et al., 2021].

Object tracking also has found a number of useful applications in the agricultural industry. Produce counting for crop yield prediction is one of such applications for a wide variety of crops [van Klompenburg et al., 2020]. Crop yield prediction is a common and essential pre-harvest practice among large farms. This is a highly challenging problem in precision agriculture which requires the use of several datasets as it depends on many different factors such as weather conditions, type of soil, use of fertiliser, and seed variety [Erkan and Dogan, 2019]. An accurate crop yield prediction solution can help farmers to make informed decisions on tasks such as planting, timely application of chemicals, precision irrigation scheduling, harvesting and labour management [Liakos et al., 2018].

Tracking is also an important task to uniquely identifying and mapping crops. Maps of crops are useful to prepare an inventory of what was grown in certain areas and when. This serves the purpose of collecting crop production statistics, facilitating crop rotation records, mapping soil productivity, crop yield prediction, assessment of crop damage due to storms and drought, among other applications [Duckett et al., 2018, Zhang and Karkee, 2021].

5.2.1 Multiple Object Tracking

In Multiple Object Tracking (MOT), as the name implies, is necessary to handle multiple objects simultaneously as well as tracking the objects of interest for a long number of frames. Two main variants of tracking algorithms exist [Luo et al., 2021], namely, *detection-based tracking* (DBT) where an object detector is applied to each frame, and *detection-free tracking* (DFT) in which the knowledge about the objects to track is restricted by some initial knowledge*, e.g.*, bounding boxes in the first frame, possibly selected by users or by some specific criteria to correctly identify the object of interest to track. DBT is currently considered the most practically useful and the most actively researched [You et al., 2019, Joshi and Mewada, 2020].

An ideal object tracking algorithm will:

- Only require the object detector once
- Be able to handle when the tracked object is occluded or moves outside the boundaries of the frame
- Be able to recover objects lost between frames

In DBT, tracking is divided into two steps: the first step is to apply object detection to each frame or key frames. The second step is to associate these detections to tracks. The most notable drawback of DBT may lie in the performance of the detector used, *i.e.*, detection rate and the number of false positives the detector can produce. In theory, a good tracker should be able to handle these flaws. It should fill gaps in detections by propagating information from neighbouring frames, and it should be able to filter false positives detections also based on the information from other frames. However, in practice this is not always the case as the increment of true detections also rises the number false positives [Ravindran et al., 2021].

The output of an object tracking algorithm is a set of trajectories, *i.e*, a set of bounding box coordinates for all objects detected in the sequence of frames. DBT usually involves the following steps:

Detection: the algorithm identifies the object's location by creating a bounding box around it.

Labelling: a unique identification is assigned to each tracked object, making it possible to identify unique objects.

Tracking: the detected objects are followed as they move through frames keeping the assignment of unique labels.

5.3 Detection vs. Tracking

The fundamental difference between *detection* and *tracking* is the use of *dynamics*, *i.e.*, *detection* independently locates the object of interest in each input frame based on a model of the object. Whereas *tracking* predicts the object's location in the next frame using estimated dynamics (e.g., direction, speed, acceleration). The object's state can then be updated based on new acquired observations commonly modelled according to the problem at hand.

For instance, in the first input frame the object's position is located. Then, an estimate of the object's velocity and some prior knowledge about the direction the object is moving to can be used. This allows the method to make a prediction of where the object will be in the next frame. Then, new observations are collected, so the estimate of where the object is can be improved based on both the prediction and the new observation. As a result, an estimate of the object's velocity is available which can then be used to make a new prediction, then detect, then predict again, and so on and so forth for as many iterations as needed.

5.4 Tracking With Dynamics

Given a model of expected motion, the goal is to predict where the objects of interest will occur in the next frame, even before examining that frame. The goals of tracking can be summarised as follows :

1. Restrict the search by doing less work when looking for the objects of interest.
2. Get improved estimates as predictions are made based on dynamics and measurements.

Along with these goals, some necessary assumptions are also made. Firstly, the problem at hand can be seen as some form of continuous (modelled) motion, *i.e.*, objects of interest do not disappear and reappear in different places in the scene. Secondly, the camera does not move instantly to a new viewpoint. And thirdly, there is a gradual change in pose between camera and scene.

5.4.1 Tracking as Inference

Two fundamental components must be defined before the tracking process can be carried out, namely:

- Hidden state (X): This is the state of the object, *i.e.*, the set parameters describing the object of interest.
- Measurement (Y): Set of observations, usually noisy, of the underlying state.

At each time step t, the current state changes from X_{t-1} to X_t and a new set of observations Y_t is collected. The goal is to estimate the distribution of state X_t given all the observations seen so far and the knowledge about dynamics of state transitions. The steps involved in tracking can be readily summarised in the following list [Doucet, 2001]:

- *Prediction*: Determine the next state of the object of interest given past measurements or observations: $P\left(X_t \mid Y_0 = y_0, \ldots, y_{t-1} = Y_{t-1}\right)$
- *Correction*: Compute an updated estimate of the state from prediction and observations: $P\left(X_t \mid Y_0 = y_0, \ldots, y_{t-1} = Y_{t-1}, Y_t = y_t\right)$
- *Tracking*: The process of propagating this posterior distribution of state given the observations across time.

We can also introduce some simplifying assumptions:

- Only the immediate past matters to predict a new posterior distribution of the current state: $P\left(X_t \mid X_0, \ldots, X_{t-1}\right) = P\left(X_t \mid X_{t-1}\right)$
- Measurements or observations depend only on the current state, therefore past observations can be deemed irrelevant: $P\left(Y_t \mid X_0, Y_0, \ldots, X_{t-1}, Y_{t-1}, X_t\right) = P\left(Y_t \mid X_t\right)$

Let us consider a base case and assume we have some initial prior that predicts state in the absence of any evidence $P(X_0)$. At the first frame, a measurement is taken or an observation is collected, then the estimate is corrected given the value of $Y_0 = y_0$. The process then iterates such that, given a corrected estimate for frame t, a new prediction and a new correction based on current observations are made for frame $t + 1$. In this process, both the state and observations can be implemented in *Sequential Monte Carlo* methods or *Particle Filters* [Liu and Chen, 1998]. The filtering problem consists of estimating the internal states in dynamical systems when partial observations are made, and random perturbations are present in the sensors of the system.

5.5 Particle Filters

Particle filters are a set of algorithms that can be used in Bayesian statistical inference to estimate the internal states in dynamical systems when partial observations are made and random perturbations are present in the state measurements [Del Moral and Doucet, 2014]. The idea of the particle filter is based on Monte Carlo methods [Doucet, 2001] and can be used in any form of state space model. The term *particle filter* was first used by [Del Moral, 1996] in reference to mean-field interacting particle methods used in fluid mechanics. The fundamental idea is to compute the posterior distributions of the states given some noisy and partial observations.

Particle filters use a set of particles or *samples* to represent the posterior distribution of some stochastic process and their predictions are made based on probabilities, *i.e*, a likelihood value is assigned to each particle that represents the probability of that particle being sampled from the probability density function (PDF). Although the PDF in the algorithm is only an approximation of the real distribution, in practice, the random quantity can satisfy a Gaussian distribution when solving a nonlinear filtering problem, and can even express other distributions than just the Gaussian model. It also has a stronger ability to model the nonlinear characteristics of variable parameters. Thus, particle filtering can accurately express the posterior probability distribution based on observation measurements and control quantities [Speekenbrink, 2016].

A particle filter is also well known to enable robust object tracking [Elfring et al., 2021]. The goal is to track the state sequence x_k using estimated control dynamics, where x is a state vector at a discrete time step $k \in \mathbb{N}$. For estimating the state, is necessary to have a *motion model* encoding some prior knowledge on how x_k is expected to move from frame to frame. It is also required to have an *observation model* that relates environment observations to x_k.

The motion model indicates how the state changes over time under the dynamic variables: $x_k = f_k(x_{k-1}, u_k, \varepsilon_k)$. Here f_k is a function that associates x between time steps $k-1$ and k using a deterministic motion input u_k and a model noise ε_k representing uncertainties, often Gaussian, associated to the variables.

The observation model $z_k = h_k(Zx_k, \xi_k)$ is a function h_k that associates the state x_k with an expected set of observations Z and a model ξ_k representing observation noise. At each time step k, the current state changes from x_{k-1} to x_k and a new set of observations z_k is collected. The goal is to estimate the distribution of state x_k given all the observations seen so far and the knowledge about state dynamics. The steps involved in tracking broccoli heads based on a particle filter can be written as follows:

Prediction: predict a distribution of the next state of each broccoli head given past observations and dynamic actions encoded in a motion model: $p\left(x_k \mid x_{k-1}, u_k\right) = p\left(x_k \mid x_{1:k-1}, z_{1:k-1}, u_{1:k}\right)$. The *motion model* is described in section 5.5.1.

Observation: compute an updated estimate of the state from predictions and observations: $p\left(z_k \mid x_k\right) = p\left(x_k \mid x_{1:k-1}, z_{1:k-1}, u_{1:k}\right)$. For each propagated particle, the likelihood $p\left(x_k \mid z_k\right) \propto p\left(z_k \mid x_k\right) p\left(x_{k-1}\right)$ is measured using an observation model. After computing the likelihood of each particle, the likelihoods are normalised and treated as weights. The *observation model* is described in section 5.5.2.

Resampling: The particles are resampled using Sequential Importance Resampling (SIR) [Doucet and Johansen, 2009] to avoid loss of sample diversity, and then propagated to a new state distribution given the observations collected across time.

This process of prediction, observation and resampling repeats itself for as many iterations as needed. A particle filter has several advantages [Speekenbrink, 2016]:

- It is relatively simple to implement.
- It can use adaptive computation depending on the available resources (both

time and CPU).

- It can exhibit adaptive convergence rate depending on the requirements for precision and time, by changing, for instance, the initial noise level.

Conversely, there are also problems with a particle filter algorithm [Elfring et al., 2021]:

- A large number of samples are needed to closely approximate the posterior probability density of the system. The more complex the environment, the more samples are needed to describe the posterior probability distribution. However, an adaptive sampling strategy or other heuristics such as a suitable size can effectively reduce the number of samples needed.
- The resampling phase can result in loss of sample diversity, leading to sample degradation. However, SIR and similar strategies offer a mechanism to circumvent this.
- Particle filters do not have a strict proof of convergence. In theory, we have traded a *proof for local convergence* with a global search method which has no proof of global convergence but is at least guaranteed not to get stuck at local optima.

Particle filter methods are more suitable for complex non-linear tracking scenarios, like many scenarios in agriculture. The task, however, should be properly selected as Kalman Filter based solutions may be more adequate. Nevertheless, Kalman Filter are known not to converge to the correct posterior in the presence of non-Gaussian distributions [Mandel et al., 2012]. In addition, Particle filters are good at modelling how the object moves between frames, providing an estimate of where the object is expected to appear in subsequent frames. In contrast, Kalman Filter based methods perform tracking without using dynamical information. The update of state information is from the critical role of associating data of the tracked objects in adjacent frames. For some applications, Particle Filter methods and Kalman Filter based solutions can be mixed to deal with state and parameter estimation issues [Shariati et al., 2019], or to avoid occlusion problems [Lan et al., 2020], or to handle the dimensionality and nonlinearity issues affecting both techniques [Grooms and Robinson, 2021].

5.5.1 Motion Model

To propagate particles, a simple linear weighted consensus model can be used to predict the broccoli head's position in the current frame based on its position in the previous frame:

$$x_{j,i} = \left(\sum x_{j,i-1} \times w_{x_{j,i-1}}\right) - k_j^* + \epsilon \tag{5.1}$$

where k_j^* is the most likely particle for trajectory j, and ϵ is the added Gaussian noise. This model simply would give an increment to each particle plus some added noise based on current states. However, in our datasets, broccoli heads change their state based on the harvesting vehicle speed and the frame rate at which the data was collected (see Section 3.1.1.1). For this reason, it was deemed necessary to monitor broccoli crops motion change so it can be modelled more accurately. Figure 5.1 shows the location evolution of a single broccoli head collected from a sample sequence of 30 frames.

The dispersion in the $[x]$ and $[z]$ axes in Figure 5.1 is fairly consistent but for $[y]$ is 2.6 times higher. This is expected as broccoli heads move along the $[y]$ axis while the other two axes get very small variations. This can be used to set a motion increment at speed v for all three axes and also set the mean and dispersion values as a model for added Gaussian noise:

$$x_{j,i}^{[x]} = x_{j,i-1}^{[x]} + \Delta v_i^{[x]} + \epsilon \tag{5.2}$$

$$v_i^{[x]} = v_{i-1}^{[x]} + \xi \tag{5.3}$$

$$x_{j,i}^{[y]} = x_{j,i-1}^{[y]} + \Delta v_i^{[y]} + \epsilon \tag{5.4}$$

$$v_i^{[y]} = v_{i-1}^{[y]} + \xi \tag{5.5}$$

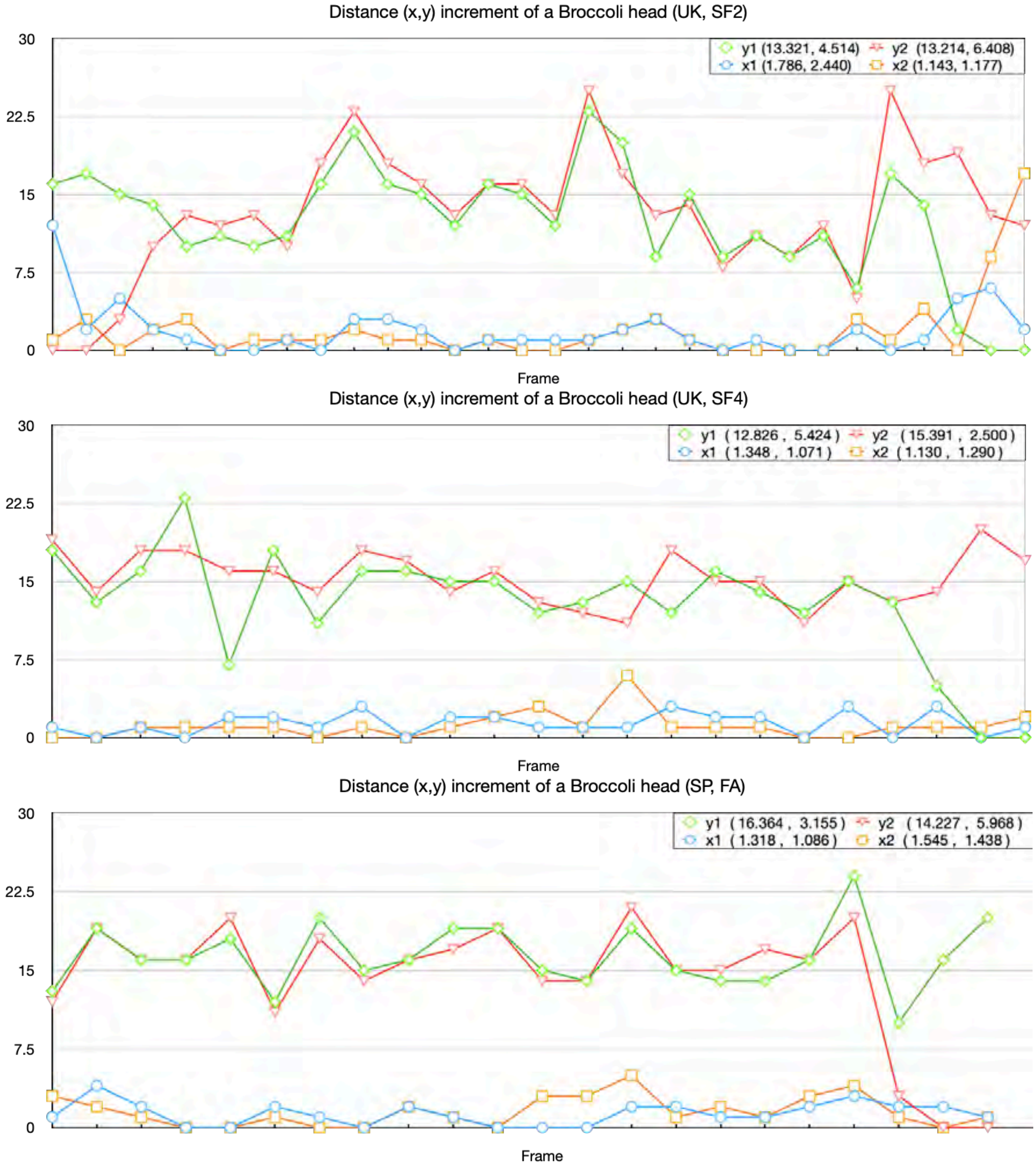


Figure 5.1: Distance increment on the x, y axes for one broccoli head visible for a sample set of 30 consecutive frames from both the UK and the Spain datasets. The values between parenthesis are the mean and the standard deviation of the shown increments (best seen in colour).

This would mean that the propagated particles would be searching for the next best prediction in the x, y space. This space can be easily translated and implemented into the space of indices used by PCL [Cousins and Rusu, 2011] to store each data point in the point cloud and to propagate particles modelled by bounding boxes formed by two point indices $(I_j, J_{j,i})$ for each particle $x_{j,i}$ that belong to a trajectory j. Therefore, the final motion model is:

$$x_{j,i}^{[I]} = x_{j,i-1}^{[I]} + \Delta v_i^{[I]} + \epsilon \tag{5.6}$$

$$v_i^{[I]} = v_{i-1}^{[I]} + \xi \tag{5.7}$$

$$x_{j,i}^{[J]} = x_{j,i-1}^{[J]} + \Delta v_i^{[J]} + \epsilon \tag{5.8}$$

$$v_i^{[J]} = v_{i-1}^{[J]} + \xi \tag{5.9}$$

Here Δ is the offset of the index point displacement for which the next prediction is estimated. This value in the space of point indices is equal to the width of the input point clouds and the initial speed v is set to twice the frame rate plus the added Gaussian noise ξ.

5.5.2 Observation Model

The observation model, on the basis of which the a posterior probabilities are computed, builds the core of any particle filter. In this thesis work, the observation model is a similarity function based on a Chi-square histogram distance χ^2 to find similarity between two 3D feature descriptor histograms and to compute particle likelihoods. Chi-square distance is one of the distance measures that can be used as a measure of similarity between two descriptor histograms and has been widely used in various applications such as image retrieval and object classification [Pele and Werman, 2010].

In this chapter, we use the simple method of measuring the inverse of χ^2 from the reference model to the nearest corresponding particle as follows:

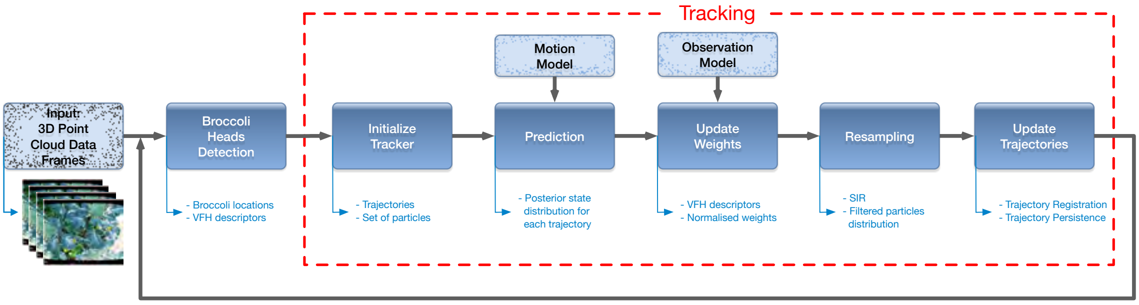


Figure 5.2: A procedure flow of the framework based on detection and tracking of broccoli heads (best seen in colour).

$$w^*_{x_{j,i}} = \frac{1}{1 + \chi^2\left(H_{x_{j,i}}, H_{r_j}\right)} \tag{5.10}$$

$$\chi^2\left(H_{x_{j,i}}, H_{r_j}\right) = \sum_i^n \frac{\left(b_{x_{j,i}} - b_{r_j}\right)^2}{\left(b_{x_{j,i}} + b_{r_j}\right)} \tag{5.11}$$

in which b are the bins of the compared histograms, $H_{x_{j,i}}$ is the descriptor histogram for particle $x_{j,i}$, and H_{r_j} is the descriptor histogram of the reference model for trajectory j and frame i. When an object has been tracked for a long time, its appearance will change, so the track weight is updated for every frame in which the track is confirmed.

5.6 Experimental Setup

In our evaluation we use the same datasets used in the experiments first presented in [Kusumam et al., 2017]. Details of these datasets were also extended in Section 3.1.1 of Chapter 3. Similarly, the experimental evaluation uses the broccoli head detectors previously discussed in Chapter 3 [Montes et al., 2020]. The general approach of broccoli head detection and tracking used throughout this chapter is depicted in the diagram shown in Figure 5.2.

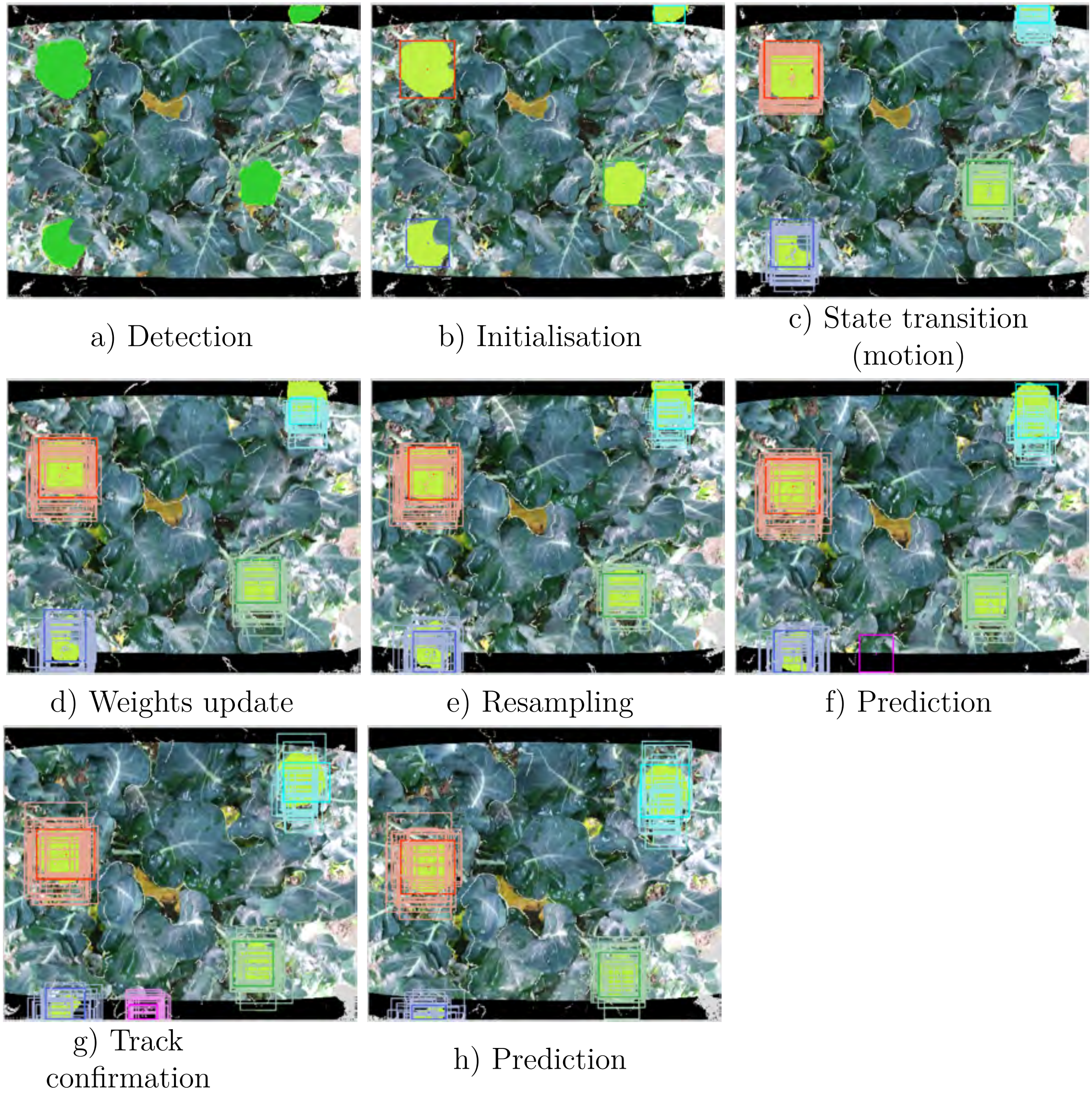
Initially, a detector is used to find all broccoli heads in the first frame and to create initial trajectories accordingly. The broccoli head state x is modelled as a set of weighted particles representing a distribution of the returned locations in the frame. This simple model will suffice as broccoli head orientations with respect to the sensor remain fairly constant while they remain visible on the scene, and the small changes perceived in orientations have a negligible impact on the tracker.

The initial set of particles is at the centre of each broccoli head the first time it is detected. The filter propagates particles from one frame to the next using the motion model. We use a standard dynamical model defined as: $x_k = x_{k-1} + \Delta * v_k + \varepsilon$; where Δ is a constant of motion increment, $v_k = v_{k-1} + \epsilon$ is the velocity and ε and ϵ are added Gaussian noise. The state x_k can then be updated based on new acquired observations provided by the broccoli detector on each new frame.

Some particles are selected or *filtered* by assigning them a weight based on its likelihood of predicting the new state correctly. Particle likelihoods are computed using an appearance model based on a Viewpoint Feature Histogram (VFH). A VFH is a 3D feature descriptor that uses normal vector angles to represent the properties of data points (ref. Section 3.1.2.3, Chapter 3). As discussed above, we use the Chi-square similarity coefficient between the predicted state and the observed histograms to compute a particle's likelihood. The likelihoods are then normalised and treated as weights. The algorithm produces a new set of particles by resampling from the current set with probabilities proportional to their weights. Figure 5.3 shows an example of an output frame sequence of the particle filter in its various stages.

5.6.1 Algorithm

For this chapter, the uncertainty about a broccoli head state (*i.e.*, its location in the 3D point cloud frame) is represented as a set of weighted particles, each one representing a possible state. The method initialises separate trackers for each crop detected, and the initial distribution of the particles is on the same location of the broccoli head the first time it is detected. Each tracker propagates particles from frame $i-1$ to frame i using the *motion model.* The trackers then compute *weights* for each propagated particle using the *appearance (observation) model.* Both models have been designed according to the task of tracking broccoli heads in 3D point cloud frames. The entire process is reflected on the steps shown in Algorithm 5.1.


a) Detection

b) Initialisation

c) State transition (motion)

d) Weights update

e) Resampling

f) Prediction

g) Track confirmation

h) Prediction

Figure 5.3: Sequence of image samples of the detect-and-track process of broccoli heads (best seen in colour). **a)** First, the inputs of the particle filter are the crop locations and their corresponding feature vectors. **b)** A new trajectory for each crop detected is created and its initial set of particles is at the centre when it is first detected. **c)** Each trajectory then propagates particles (seen as bounding boxes) to the next frame using the dynamical model. **d)** Particles weights are then computed using the appearance model and the crop's feature vector as observations. **e)** The particles are resampled (filtered) using SIR to avoid lost of particle diversity. **f)** A new broccoli head prediction -the best particle- is then determined. However, faulty observations, *i.e.*, false positive results from the detector, cause the tracker to initialise a new trajectory (shown in violet). **g)** These false trajectories are either confirmed (to handle misdetections) or eliminated in the next iteration. **g)** Finally, a new prediction of broccoli heads is made and the process continues for the rest of the input frames.

Algorithm 5.1 Detect-and-Track algorithm followed in this chapter for the task of tracking broccoli heads in 3D point cloud frames.

```
1. Acquire the input frame sequence $S$ of planted broccoli crops.
2. In the first frame $s_0$ of $S$, detect the broccoli heads and let $x_{j,0} = (I_j, J_{j,i}), j \in 1...N$ be the bottom left and the top right positions of the points that form a bounding box around each detection.
3. Initialise broccoli head trackers $T_j, j \in 1...N$ with initial positions at $x_{j,0}$.
4. Initialise occlusion count $O_j$ for each tracker $j$ to 0.
5. Initialise the similarity model $z_{j,0}$ for each tracker $j$ from the region around $x_{i,0}$.
6. For each subsequent frame $s_i$ of $S$,
   a) For each existing tracker $T_j$:
      i. use the *motion model* to predict the distribution $p(x_{j,i}|x_{j,i-1})$ over locations for head $j$ in frame $i$ to create a set of candidate particles $x^k_{j,i}, k \in 1...K$.
      ii. compute likelihood $p(z^k_{j,i}|x^k_{j,i})$ and the similarity distance $d^k_{j,i}$ for each particle $k$ using the *observation model.*
      iii. resample the particles according to their likelihood (*importance sampling*). Let $k^*_j$ be the index of the most likely particle for tracker $j$.
   b) Perform confirmation by data association:
      i. Run the broccoli head detector on frame $s_i$ and let $x_l, l \in 1...M$ be the bounding box indices for each new detection.
      ii. For each tracker $T_j$, find the detection $x_l$ closest to $x^{k^*_j}_{j,i}$ . If found, consider the location as a head and reset $O_j$ to 0; otherwise, increment $O_j$.
      iii. Initialise a new tracker for each detection not associated with a tracker in the previous step.
      iv. Delete each tracker $T_j$ that either has occlusion count $O_j$ greater than a track survival threshold $O_{top}$, or has moved outside the frame.
```

5.6.1.1 Trajectory Confirmation

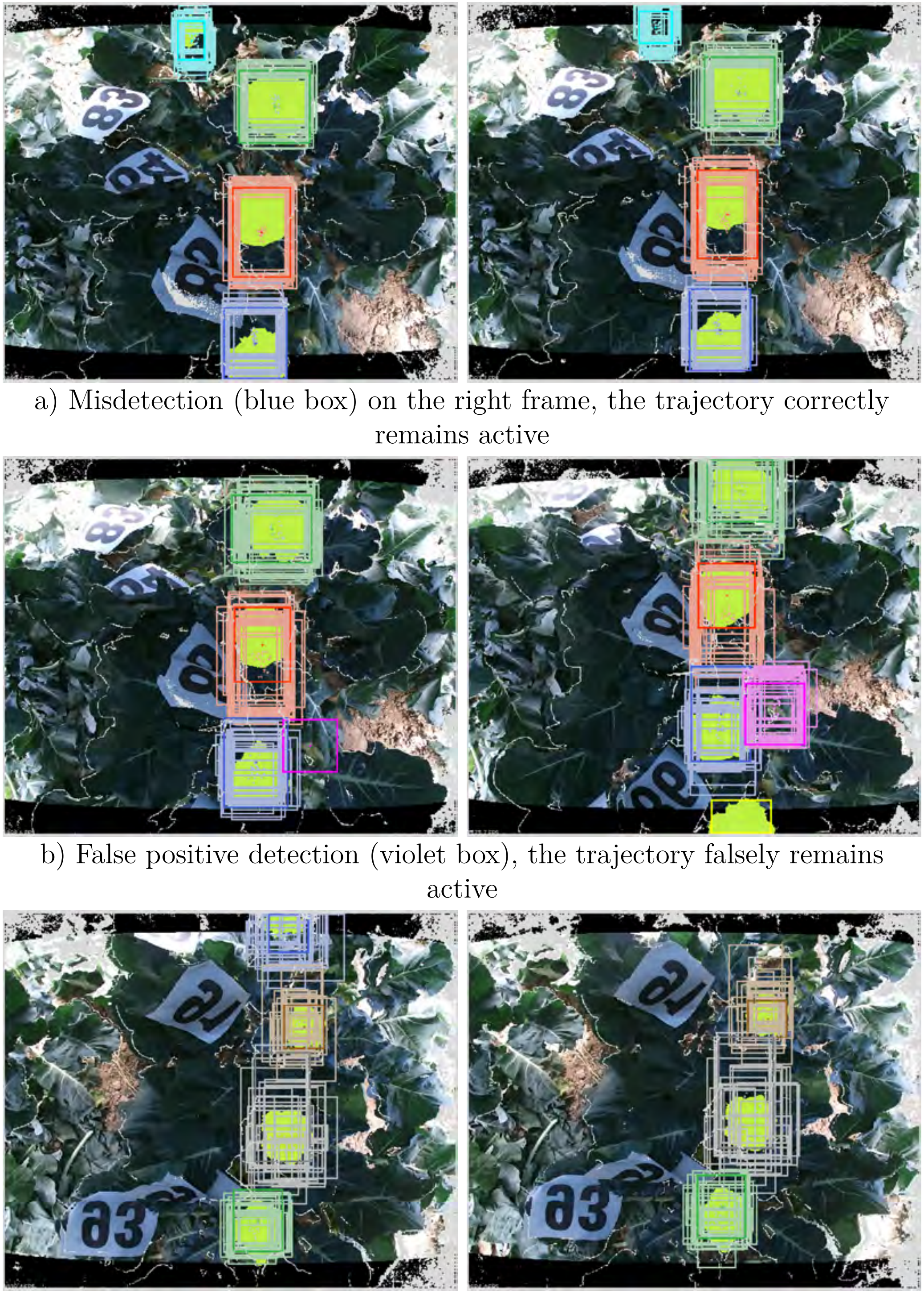
When the detector's response is accurate, it can guide the tracker fairly well using simple data association policies. Unfortunately, detectors are not fully reliable and a trade-off between true positive and false positive rates is common. However, increasing true detections also increases false positive rates. When false detections occur, simple rules often misguide the tracker and perform poorly. This problem can be alleviated by introducing a confirmation step that performs data association and handles missing detections.

For the experiments presented in this chapter, on each frame the detector is first used to confirm the prediction result for each trajectory. If no trajectory has been associated to a detection, its posterior state is simply predicted without confirmation using the motion model. This policy is essential and highly useful when it comes to handle both faulty observations and occlusions. Meanwhile, trajectories that have not been confirmed for O_{top} frames are terminated, and any trajectory either near or beyond the border is also removed to avoid predictions outside the current frame.

In our experimental evaluation, this method provides an effective tracking of broccoli heads in sequences of 3D frames and improves accuracy by reducing false positive predictions without losing tracks of crops not detected for short periods of time while preserving high detection rates. Nevertheless, if either the detector keeps reporting false positive detections, or the data association still finds sufficient similarity between current and posterior predictions, the corresponding trajectories will still remain active. Figure 5.4 shows the type of trajectories that needed to be confirmed by the algorithm.

5.6.2 Evaluation Metrics

Location and area size are the main parameters associated with correctly detecting and tracking each broccoli head. In our experiments, a correct state estimation made by the particle filter matches at least a 0.5 Intersection-over-Union value according to annotated data. In the analysis presented in this chapter, the common metrics *Precision*, *Recall* and *F1 Score* suffice to evaluate performance, as Precision is the fraction of true predictions among only the total positive predictions made by the system, and Recall is the number of true positive results divided by the number of all



a) Misdetection (blue box) on the right frame, the trajectory correctly remains active

b) False positive detection (violet box), the trajectory falsely remains active

c) Trajectory correctly eliminated near the top frame edge

Figure 5.4: Selected frame samples of the trajectory confirmation step (best seen in colour). **a)** Even when a misdetection occurs, the trajectory remains active and keeps tracking the broccoli head target. **b)** A trajectory could remain active as long as the detector still reports a detection, albeit false, or the tracker determines that the posterior prediction is still sufficiently accurate. **c)** Trajectories near or beyond the border frame are immediate candidates for elimination.

observations expected to be identified as positive, both based on labelled datasets.

In comparison, F1 Score is the weighted average of both Precision and Recall; consequently, it takes both false and missing positive predictions into account and it is usually more useful when there is an unbalanced class distribution in the datasets [Saito and Rehmsmeier, 2015]. As a result, the F1 Score is chosen in this chapter as a measure of the system's accuracy. The highest possible F1 Score value indicates a perfect Precision and Recall, and a 0 value would mean that either precision or recall are also zero.

These measures, however, are known to present specific biases [Powers, 2011]. Precision, for instance, is a description of observational errors and a measure of statistical dispersion. It measures how close or dispersed positive results (including those wrongly predicted) are to each other. This means that Precision measures how well an autonomous harvester cutting tool would pick a broccoli head even in the cases where there is none in the predicted locations. Recall, on the other hand, measures the ratio of the number of broccoli positive predictions to the real number of broccoli heads. This means that Recall alone measures how well a harvester would cut only the truly positive broccoli heads, though ignoring those not predicted as positive but actually labelled as broccoli crops. For these reasons, a balance between these two metrics is preferred.

5.6.2.1 Tracking Performance

The system is evaluated with all training and testing combinations of annotated datasets collected in two different runs in the UK, labelled as UK1 and UK2, and one dataset from Spain, labelled as SP1. For experiments on the same set, we define a training and testing split of a 75% to 25% ratio. In any other case, 100% of one dataset was used for training and 100% of the other set was used for testing.

Our experimental setup includes results for the tracking task using all four broccoli heads detectors detailed in Chapter 3. As a result, Tables 5.1, 5.2, 5.3 and 5.4 show a comparative list of tracking performance results when using the different broccoli detectors and the proposed detect-and-track framework.

In our experiments, the quality of the detector has a significant impact on the tracking performance, since changing the detector can improve the overall tracking

			Train set											
			UK1				UK2				SP1			
			Pr	Rc	F1		Pr	Rc	F1		Pr	Rc	F1	
Test set	UK1	Det	98.5	96.6	97.5	0.8	97.9	92.3	95.0	1.3	96.4	88.7	92.4	1.8
		PF	98.1	98.5	98.3		96.2	96.4	96.3		93.3	95.2	94.2	
	UK2	Det	97.0	91.6	94.2	1.5	97.2	95.7	96.4	1.1	88.9	84.3	86.5	6.6
		PF	95.5	95.9	95.7		96.3	98.7	97.5		92.2	93.9	93.1	
	SP1	Det	97.6	85.1	90.9	3.5	98.9	83.9	90.8	4.4	96.1	91.7	93.8	1.5
		PF	94.5	94.3	94.4		97.8	92.6	95.2		94.3	96.3	95.3	
95.6/2.5					96.1	1.9			96.3	2.3			94.2	3.3

Table 5.1: Tracking results when using the OES detector and when the Particle Filter is added for tracking.

			UK1				UK2				SP1			
			Pr	Rc	F1		Pr	Rc	F1		Pr	Rc	F1	
Test set	UK1	Det	99.2	91.7	95.3	1.3	98.0	91.1	94.4	1.4	95.1	89.8	92.4	1.0
		PF	98.1	95.1	96.6		96.9	94.7	95.8		92.4	94.4	93.4	
	UK2	Det	97.9	88.6	93.0	2.5	98.3	90.1	94.0	2.2	94.4	90.1	92.2	1.0
		PF	96.9	94.1	95.5		96.8	95.7	96.2		91.9	94.5	93.2	
	SP1	Det	93.5	86.1	89.6	2.4	94.8	85.9	90.1	3.0	98.3	93.4	95.7	1.0
		PF	91.4	92.5	92.0		93.4	92.8	93.1		97.1	96.3	96.7	
94.7/1.8					94.7	2.1			95.0	2.2			94.4	1.0

Table 5.2: Tracking results for multiple train and test combinations of the broccoli datasets when using ORGS detector (Det) and when the Particle Filter (PF) is added for tracking. The values shown in the first cell row are the *Precision* (Pr), *Recall* (Rc) and *F1* evaluation metrics, while the values at the bottom row are the average F1 Score and the average F1 increment difference.

performance by up to 3.1% on average. The F1 scores for each dataset combination indicate an improved performance from detection results in all instances. This can be seen, for example, in the results shown in Tables 5.1 and 5.4 where the best detectors lead to the best tracking accuracy. In these results, an average increment in the F1 score of 95.0% and 95.6% is achieved on average when the particle filter is added to both the OES and the EAC detection pipelines.

The method presented in this chapter uses the output of an object detector as a cue to identify instances of broccoli heads, but is robust to False Positive detections, low

			Train set											
			UK1				UK2				SP1			
			Pr	Rc	F1		Pr	Rc	F1		Pr	Rc	F1	
Test set	UK1	Det	98.4	95.1	96.7	0.5	98.3	91.7	94.9	1.3	98.1	88.3	93.0	2.1
		PF	96.3	98.1	97.2		97.7	94.7	96.2		96.1	94.1	95.1	
	UK2	Det	98.2	90.9	94.4	1.6	97.2	96.8	97.0	0.1	97.0	91.3	94.1	1.1
		PF	96.3	95.7	96.0		95.4	98.8	97.1		95.3	95.1	95.2	
	SP1	Det	91.4	83.5	87.3	3.7	89.0	83.9	86.4	2.8	96.0	89.6	92.7	2.0
		PF	89.1	92.8	91.0		85.3	93.0	89.2		93.9	95.4	94.7	
94.6/1.7					94.7	1.9			94.2	1.4			95.0	1.7

Table 5.3: Tracking results when using the FEC detector and when the Particle Filter is added for tracking.

			UK1				UK2				SP1			
			Pr	Rc	F1		Pr	Rc	F1		Pr	Rc	F1	
Test set	UK1	Det	98.4	92.1	95.1	1.4	98.4	89.5	93.7	1.8	97.0	83.4	89.7	4.0
		PF	97.6	95.4	96.5		97.9	93.2	95.5		96.1	91.4	93.7	
	UK2	Det	97.0	86.5	91.4	2.8	98.3	92.5	95.3	2.1	97.4	86.0	91.3	3.5
		PF	96.1	92.5	94.2		97.2	97.6	97.4		96.7	93.0	94.8	
	SP1	Det	96.1	81.6	88.3	5.0	95.2	80.4	87.2	5.2	97.8	92.1	94.9	2.1
		PF	95.2	91.4	93.3		94.9	90.0	92.4		96.8	97.2	97.0	
95.0/3.1					94.7	3.1			95.1	3.0			95.2	3.2

Table 5.4: Tracking results for multiple train and test combinations of the broccoli datasets when using EAC detector (Det) and when the Particle Filter (PF) is added for tracking. The values shown in the first cell row are the *Precision* (Pr), *Recall* (Rc) and *F1* evaluation metrics, while the values at the bottom row are the average F1 Score and the average F1 increment difference.

bounding box localisation and occlusions.

The performance of any particle filter scale with the amount of resources available, such as computing power, which may lead to a larger number of samples or particles representing the posterior state distribution, which, in turn, will lead to better tracking results. By increasing the number of particles, the tracking estimate can be shown to converge to the exact solution [Doucet, 2001, Doucet and Johansen, 2009]. However, the choice of sample size is usually determined based on a trade-off between computational cost and the variance of the tracking performance. As the

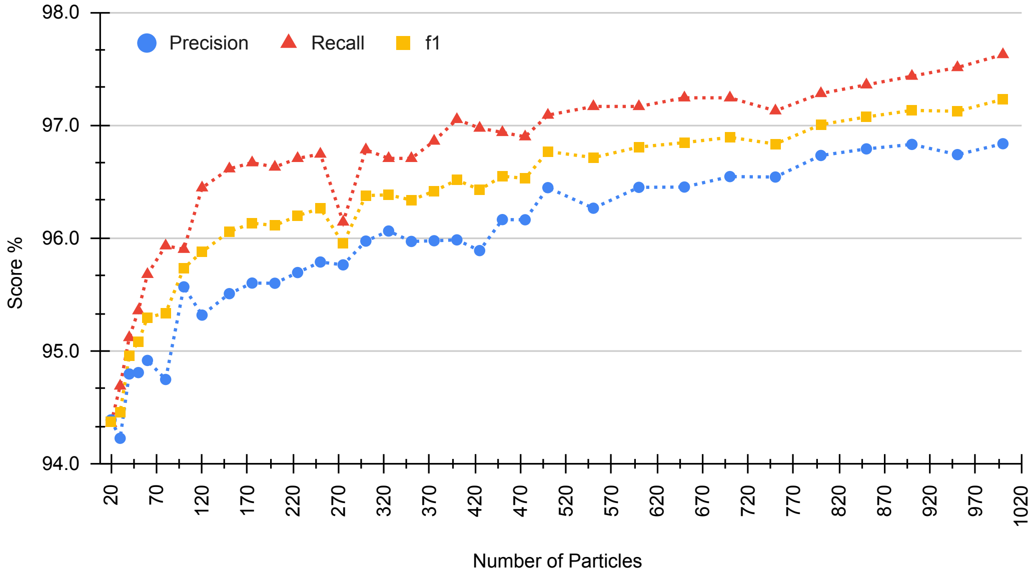


Figure 5.5: The graph shows the relationship between the number of particles used by the particle filter and the tracking performance on the UK dataset (best seen in colour).

number of particles or sample size increases, the former also increases; while the latter takes longer to exhibit substantial increments. This can be seen in Figure 5.5 where the relationship between the number of samples used in the particle filter and the tracking performance shows a F1 score around 96% for a sample size in the vicinity of 120 particles. The F1 Score does tend to rise as the sample size gets larger, but better performance entails longer executions times on the hardware used in our experiments.

5.6.2.2 Running Times

A particle filter performance is satisfactory only when the set of particles is sufficiently large to represent the state distributions that are being estimated. However, the number of particles used have an impact in the running time of the tracker.

It would be relatively simple to increase the overall accuracy by allowing a larger number of particles to represent the posterior state distribution. This, however, would be at the expense of runtime performance as a gradual increment in the number of particles also increases running times. For some tracking applications, running times may not be relevant, but for a autonomous selective harvester operating in open farm fields, a real-time operation is essential.

Figure 5.6 shows how the relationship between the number of particles used in the filter and the tracking results. The graph shows that the F1 Score metric starts to rise steeply between 96.0 and 96.3%. By looking at Figure 5.5, this F1 performance score is produced when the tracker uses under 120 particles. In our evaluation, a small set of 100 particles per trajectory was large enough to improve the performance score of the system and only added a few milliseconds to the process, still allowing the real time execution reported in [Montes et al., 2020] using the same computing hardware. Even though this is just a practical approach and not a rigorous method, this number of particles seems reasonable in the sense that increasing the number of particles further will not increase performance without costing substantially longer running times.

In general, the method discussed in this chapter is pragmatic in nature due to the lightweight models involved, which are identified as a key factor for tracking multiple broccoli crops in real-time. In practice, the process can be readily sped up by running the detector only once every few frames. This can be boosted even further by employing some basic multithreading and concurrency implementation techniques. For instance, the detector can be run on a slow thread looking for broccoli heads to track and, once those heads are properly locked on, then the tracker, running on a faster thread, can takeover. Similarly, individual particles can be executed on separated threads as a way of easily speed up the entire process.

5.7 Conclusions

The framework discussed in this chapter utilises a real-time detector, 3D feature vector histograms to model similarity appearance, and a simple track confirmation technique to keep track accuracy through fail detections. The results demonstrate that the system is capable of reliably detecting and tracking multiple instances of broccoli heads in sequences of 3D frames, as well as improving accuracy by reducing false predictions while preserving high detection rates. Our results indicate a consistent improvement of the F1 Score in all datasets combinations used for testing.

The algorithms detailed in Chapter 3 were used as detectors in the tracking frame-

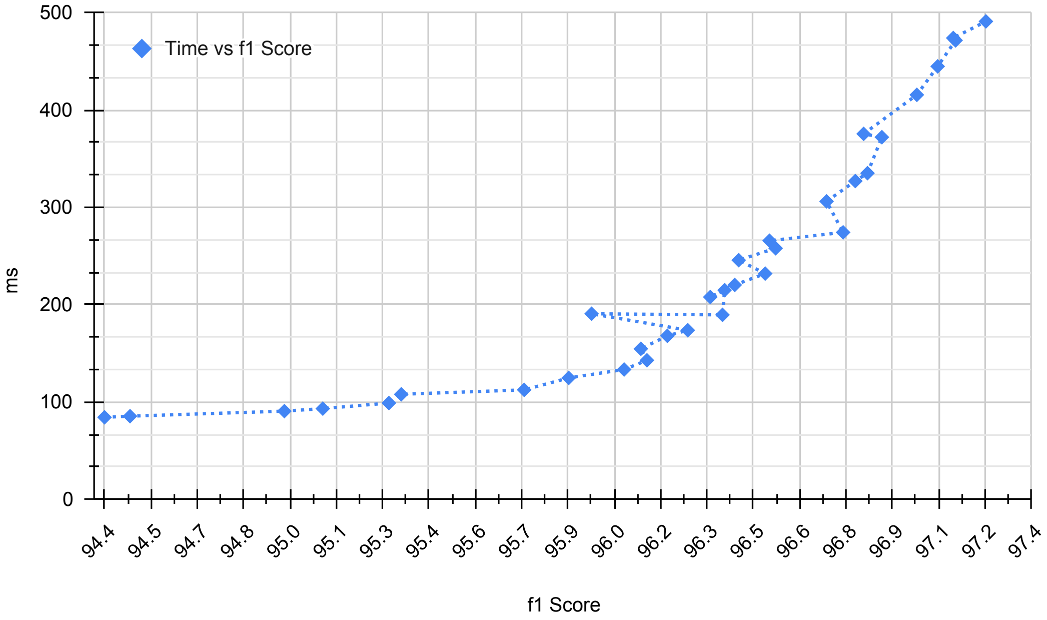


Figure 5.6: Graph showing the relationship between the number of particles used in the particle filter and the tracking results when using the OES detector (best seen in colour).

work described in this chapter. Our experimental results indicate that OES is the most suitable detector, both in terms of accuracy and speed. However, the framework also shows limitations. The detector and tracker used in the chapter's experimental setup are designed to meet real-time execution, but they also sacrifice tracking performance in some cases. One instance is the case of occlusions occurring for longer periods than those anticipated during execution, for which the broccoli heads being tracked are lost or their IDs are renewed, thus reducing the tracking effect. Nevertheless, based on the frame rate at which the datasets were collected, this problem is reduced with the framework's current settings. Even though broccoli heads do not appear to be occluded for too long, the existence of this issue cannot be disregarded and must be addressed in the immediate future research by focusing, for instance, on object re-identification to handle long term occlusions.

The implication of tracking of each broccoli head during autonomous selective harvesting is worth noting as well, considering that in real field conditions each broccoli head must be individually assessed to determine whether it is ready to be harvested or not. Other horticultural crops suitable for slaughter or mass harvesting, do not need to be individually detected by similar applications. However, tasks such as counting for crop yield estimation still need an accurate tally of the planted produce. The generic nature of the framework presented here makes it applicable to a wide range of other tasks in agriculture.

Due to the computational resources available today, it is feasible to carry out a *track-by-detect* approach, *i.e.*, tracking is performed by just detecting the crop target at very high frame rates. Such an approach, however, discards the kinematics of the object (*e.g.*, velocity, acceleration) which would help in predicting the location of the crop in subsequent frames. Tracking is about giving each target a unique identity so that objects can be observed and tracked throughout a long time interval. This is especially important for applications where trajectories must not mix, even when a tracked crop temporarily occludes another. In such scenarios, tracking is helped by having available the kinematics of the crops to track.

Crops can be tracked based solely on motion features without identifying or knowing anything about the actual crops being tracked. Thus, pure object tracking can be extremely efficient if it leverages the temporal relationship between frames.

Given that in modern state-of-the-art applications currently deployed in farm fields, a full harvesting cycle time, *i.e*, from perception to harvesting, takes between 30 to 40 seconds to complete [Kootstra et al., 2021, Oliveira et al., 2021], efficient methods are essential for real-time applications in agriculture. The framework detailed in this chapter uses the simple and lightweight models of a particle filter aiming, precisely, to meet the real-time execution required by selective harvesting operations in real farm field conditions, which emphasises the role played by the detector within the framework: the tracking performance is highly dependant on the detector performance. On a modern CPU, the system is able to efficiently tracking broccoli locations, thus enabling a fast execution with implications for other real time detection and tracking applications. Current state-of-the-art results show, nonetheless, that modern Deep Learning techniques are deemed higher rank alternatives, as they have become the method of choice for many detection and classification problems. They can be integrated as part of the particle filtering based tracking system or as a stand-alone tracking framework, thus reducing the dependence on the detector and improving the overall tracking performance [Joshi and Mewada, 2020, Zhang et al., 2021].

6 Summary, Overall Conclusions and Future Work

The agricultural industry is facing huge challenges, from critical labour shortages, rising costs of supplies, changes in consumer preferences, sustainability issues, political pressures, migration dynamics, and now a worldwide pandemic and an armed conflict in Europe. Despite all these challenges, the industry has seen notable technological advancements, especially over the last three decades. In current practice, however, harvesting broccoli and other high-value crops still relies on human labour to pick the produce, thus increasing harvesting expenses and other operation costs. These high costs are a major motivation driving the development of autonomous selective harvesters for high-value agricultural products.

Autonomous robotic harvesting has many benefits over manual harvesting, such as processing the produce in shorter periods of time, higher quality of crops, and the reduction of human labour. Automated harvesting also enable new functionality based on sensing abilities absent in humans or not possible to perform at a similar accuracy, consistency and cost. These and other potential benefits have driven scores of research and commercial projects aimed at developing agricultural robot solutions to automatically harvest broccoli, fruits and other vegetables for fresh consumption. The challenge is to quickly and precisely identify the broccoli heads ready to be cut in real farm field conditions. The key behind this challenge lies in precise and fast machine learning methods capable of dealing with this complex environment.

This thesis has demonstrated the development of machine learning algorithms for detecting and tracking broccoli heads from 3D point clouds in real-time that could be applied in an autonomous robotic broccoli harvester.

6.1 Summary and General Conclusions

Detecting crops of broccoli plants from 3D data at real-time execution speeds involves extracting complex features either by carefully engineering them or by automatically synthesised them to train classifiers within a processing pipeline to effectively output the location of the broccoli heads. This PhD dissertation investigated the applicability of 3D-based machine learning algorithms and the level of precision that the methods should have in order to fulfil the accuracy needed for autonomous robotic harvesting applications. The goal throughout this research work has been that an automated selective broccoli harvester could be built using a competitively priced imaging system able to deliver the required levels of accuracy, reliability, and scalability.

To address the objective of this research, four clustering methods that operate at high frame rates were developed and analysed for this thesis. These methods achieve high performance in detection and time execution by exploiting the organised structure of the 3D data collected with low-cost RGB-D sensors. In similar fashion, a technique based on a Convolutional Neural Network (CNN) architecture was implemented and applied to organised 3D point clouds for broccoli heads detection and segmentation. This method achieves an even higher inference time at speeds of 50∼60 frames per second. The execution times of these methods make them applicable for autonomous robotic harvesting and other farming operations. The dissertation also presented a tracking method of broccoli heads based on a Particle Filter, that combines a broccoli detector with the particle filter to track multiple crops in a sequence of 3D data frames. Crop detection and tracking are together an important part of autonomous selective harvesting as they can play a key role in the accuracy of the harvester's grasping and cutting system. These two tasks should be part of an unified system towards improving generalisation and performance of automated selective harvesters to uniquely determine the correct location of the target crop.

All the algorithms described in this work have been implemented and tested on data from two broccoli varieties captured in planted farm fields under different weather conditions. The experimental results were carried out on all datasets combinations to allow the generalisation performance analysis of the algorithms and their shortcomings. Similarly, the contributions presented in this dissertation have been evaluated as part of complete pipeline systems designed to be integrated into an autonomous selective robotic harvester of broccoli crops.

Feature-based Detection of Broccoli Heads The first set of machine learning algorithms developed for this dissertation were based on a series of strategies based on pre-designed features to cluster 3D points of broccoli plants. Classification processing speed and training time are two factors that make these 3D features and conventional machine learning methods still be relevant in the era of deep learning algorithms. The classification results have shown in Chapter 3 that the sensor distance used to collect the datasets produces a high broccoli head detection rate and suggests an appropriate hardware setup for the robotic selective harvester. The results show that the Spain dataset is more difficult than the UK one, mainly due to the high number of occlusions of the broccoli heads. However, cross-validation of both datasets indicate a high generalisation performance of the pipeline under different field conditions for the two broccoli varieties.

Comparative experimental results also show that the methods presented in Chapter 3 achieved both high classification performance and real-time execution against recent approaches for broccoli detection available in the literature, either based on the Euclidean proximity of 3D points when tested on the same datasets, or based on conventional machine vision methods. These results are also comparable with the most recent deep learning models, although they still present some generalisation shortcomings. The clustering strategies implemented by all clustering algorithms involve a trade-off between area segmented and detection accuracy, as the size of the clusters extracted provides enough information for harvesting the most marketable heads.

The evaluation performance shows that the algorithms demonstrate the required detection accuracy and real-time performance needed for autonomous robotic harvesting applications. The FEC, EAC, ORGS and OES clustering algorithms achieved processing frame rates of 9.39, 10.99, 10.38 and 14.56 fps, respectively, running on a modern commercial desktop computer. This processing time improves on previous research results [Blok et al., 2016, Kusumam et al., 2017, Blok et al., 2020, Blok et al., 2021]. Similarly, the OES, FEC, EAC and ORGS algorithms achieved an average Precision of 98.1%, 96.0%, 97.4%, and 96.9%, as well as an average Recall metric of 91.5%, 90.6%, 88.1%, and 90.3%, respectively for all datasets combinations. All these results have been listed and compared with state-of-the-art methods in Table 3.2 in Chapter 3.

Deep Learning-based Detection and Segmentation of Broccoli Heads Current progress in deep learning methods have prompted this research work to demonstrate the use of an efficient and effective approach using 3D information for detection and segmentation of broccoli heads based on a Convolutional Neural Network architecture. The system achieved a high performance in terms of accuracy, segmentation, and localisation, with a better generalisation for the most difficult datasets at high processing speeds.

Unlike the set of algorithms for real-time detection of broccoli heads presented in Chapter 3, deep neural networks do not require the design of handcrafted features, as deep learning algorithms have a powerful capacity of feature expression and also have an automatic feature extraction over large datasets. In addition, deep neural networks have the property of producing higher-level features by automatically composing lower-level features. This deep learning characteristic has been further extended in Chapter 3 by directly providing additional low-level features in the form of surface normals on an organised arrangement of the input point clouds. The method achieves similar results than feature engineered clustering algorithms and better results for the most challenging datasets, while providing better segmentation of the broccoli heads, comparable instance segmentation to recent published results, better localisation, and faster inference time.

Deep learning algorithms are immediately appealing in terms of accuracy and overall performance, but they still present some challenges such as drifting and real-time processing. In particular, the CNN architecture presented here also faces some challenges intrinsic to the data. For instance, differences in size of the broccoli heads within the datasets lead to missed detections, especially on the top and bottom boundaries of the point cloud frames, where a large number of noisy data points appear and broccoli head size varies the most. Dealing with incomplete information due to occlusions is still a big challenge when operating in complex planted fields environments, although recent results have shown promising advancements on solving this problem [Blok et al., 2021].

Tracking Broccoli Heads The framework presented in this thesis uses a real-time detector, 3D feature vector histograms to model similarity appearance, and a simple track confirmation technique to keep track accuracy through fail detections. The results demonstrate that the system is capable of reliably detecting and tracking multiple instances of broccoli heads in sequences of 3D frames, as well as improving accuracy by reducing false predictions while preserving high detection rates.

However, the framework also shows limitations. The detector and tracker used in the experimental setup are designed to meet real-time execution, but they also sacrifice tracking performance in some cases. One instance is the case of occlusions occurring for longer periods than those anticipated during execution, for which the broccoli heads being tracked are lost, or their IDs had to be renewed. Even though broccoli heads do not appear to be occluded for too long, the existence of this issue cannot be disregarded and must be addressed in the immediate future research by focusing, for instance, on object re-identification to handle long term occlusions.

The implication of tracking of each broccoli head during autonomous selective harvesting is worth noting as well, considering that in real field conditions each broccoli head must be individually assessed to determine whether it is ready to be harvested or not. Other horticultural crops suitable for mass harvesting do not need to be individually detected by similar applications. However, tasks such as counting for crop yield estimation still need an accurate tally of the planted produce. The generic nature of the framework presented here makes it applicable to a wide range of other tasks in agriculture.

In general, we found that the framework detailed in Chapter 5 uses the simple and lightweight models of a particle filter aiming to meet the real-time execution required by selective harvesting operations in open farm field conditions. On a modern CPU, the system is able to efficiently tracking broccoli locations, thus enabling a fast execution with implications for other real-time detection and tracking applications.

6.2 Future work

Improvements and future directions to the work presented in this PhD dissertation are connected to both detection and tracking applications, for there is a list of open issues that still remain to be tackled for future research.

For detection, some improvements can be adopted to further enhance the generalisation of the clustering algorithms and the detection pipeline. An open research area is the design of methods to better encode the properties of the broccoli heads to achieve a more accurate clustering of 3D points. This is important in order to estimate more precisely the size of broccoli heads suitable for today's market standards. However, this might also constitute a limitation, as machine learning algorithms based on feature engineering methods involve skill and effort whose generalisation capability are either limited or enhanced by the classifier used in the pipeline and its various configuration settings as well as data augmentation and regularisation techniques to improve the overall performance of the classifier. These challenges can be addressed through some standard processes such as data normalisation, un-distortion, and data augmentation.

For the deep learning broccoli detection model, similar data normalisation and augmentation techniques need to be further researched. Recent research work found that data augmentation improved the generalisation performance of an Mask R-CNN model [Blok et al., 2020]. Also, the use of more advanced architectures could potentially improve robustness to scale and shape variation, occlusion and data diversity. However, the choice of an machine learning algorithm for detection and segmentation should be made depending on whether speed or accuracy takes priority. The major algorithms currently used in both research and industry are YOLO, SSD, Faster R-CNN, Blitznet or Mask R-CNN. All these algorithms are well performing architectures for object detection, but still fall short on an important one: speed for real-time detection. One exception is the YOLO algorithm. YOLO is faster than other detection algorithms and is used for speed and real-time applications, but it has shown difficulty detecting small objects close to each other. Nonetheless, improvements are still being made to the algorithm, which has seen four generations so far [Wang et al., 2021]. For segmentation tasks, however, other architectures are considered for future research work.

For the broccoli heads tracking framework, due to the computational resources available today, it is feasible to carry out a *track-by-detect* approach, *i.e.*, tracking is performed by detecting and matching the crop at very high frame rates. This approach, however, discards kinematic properties such as velocity and acceleration of the object, which would help in predicting the location of the crop in subsequent frames. Tracking is about giving each target a unique identity so that objects can be observed and tracked throughout a long time interval. This is especially important for applications where trajectories must not mix, even when a tracked crop temporarily occludes another. In such scenarios, tracking is helped by having available the kinematics of the crops to track. Similarly, high-value crops can be tracked based only on motion features without identifying or knowing anything about the actual crops being tracked. Thus, pure object tracking can be extremely efficient if it leverages the temporal relationship between frames.

Also, given the rather small depth variation and the sensor viewpoint, the z component was only used for feature estimation and discarded for estimating the location of each broccoli head in posterior frames of the point cloud sequence. A new motion model should be implemented in future work to include depth for estimating the broccoli head position and then compare both performance and model simplicity. The robustness of this algorithm could be improved by considering a similar probabilistic approach when estimating the 3D position and the refinement the model on every frame.

Improving tracking speed while maintaining or even improving the accuracy still needs to be further researched. Current state-of-the-art results show, nonetheless, that modern Deep Learning techniques are deemed higher rank alternatives, as they have become the method of choice for many detection and classification problems. They can be integrated as part of the particle filtering based tracking system or as a stand-alone tracking framework, thus reducing the dependence on the detector and improving the overall tracking performance.

All the studies summarised in Chapter 2 as well as the methods presented in this thesis either approach the localisation of broccoli heads, and some of them evaluate their condition of harvestable based on size. Only [García-Manso et al., 2021] conducted experiments to classify broccoli heads in one of three classes, *i.e.*, harvestable based on size and shape, immature also based on size, or wasted due to over-ripeness, defects or disease. However, it is important for a autonomous selective harvester to include other attributes to look at for broccoli quality, such as a smooth head, as a more lumpy and irregular broccoli head surface makes them unmarketable. Other attribute deformations such as discolouration (*i.e.*, mixed lime and purple colouring) as well as bracketing, which is when there are leaves coming out through the head, make the produce unsuitable for the fresh market. In current harvesting practice, all these attributes are graded in-field by human pickers. Given the performance of modern machine learning algorithms and their ability to classify objects into several classes, the detection can be made along with the localisation.

Though tested on different planted farm fields, the methods and algorithms presented in this work need to be tested on even larger datasets representing other broccoli varieties as well as various whether conditions and field environments. This will make the detection pipeline more robust. To achieve this goal, the need of public repositories of datasets for agriculture applications must be first acknowledged so that they can be available for both research and commercial applications. This could be difficult to realise at a faster pace due to privacy and copyright concerns. However, just like other research areas, some steps are incrementally be taken towards this goal.

This thesis has not addressed the construction of the two remaining critical systems of an autonomous selective robotic harvester of broccoli crops. It can, however, be concluded that the methods developed for this dissertation can be implemented in a low-cost 3D imaging hardware able to operate in real-time. Therefore, as a future work, it is intended to continue the development of a complete harvester prototype by integrating first the 3D imaging and the broccoli heads detector system. This prototype must be thoroughly tested in real field conditions before adding the grasping and cutting system, which need to be precisely synchronised in order to harvest every detected broccoli head classified as harvestable. Also, the entire planted field and the uncut broccoli heads should be mapped, so that future harvesting passes can be planned accordingly.

Although these are important issues that need to be solved in future research work, the results presented in this thesis are small steps towards the realisation of a more affordable autonomous selective broccoli harvester platform in the near future.

Bibliography

[Acuna et al., 2018] Acuna, D., Ling, H., Kar, A., and Fidler, S. (2018). Efficient Interactive Annotation of Segmentation Datasets with Polygon-RNN++. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 859–868.

[Arief et al., 2020] Arief, H. A., Arief, M., Zhang, G., Liu, Z., Bhat, M., Indahl, U. G., Tveite, H., and Zhao, D. (2020). SAnE: Smart Annotation and Evaluation Tools for Point Cloud Data. *IEEE Access*, 8:131848–131858.

[Bac et al., 2014] Bac, C. W., van Henten, E. J., Hemming, J., and Edan, Y. (2014). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics*, 31(6):888–911.

[Bachche, 2015] Bachche, S. (2015). Deliberation on design strategies of automatic harvesting systems: A survey. *Robotics*, 4(2):194–222.

[Baenas and Wagner, 2019] Baenas, N. and Wagner, A. E. (2019). Pharmacoepigenetics of Brassica-Derived Compounds. In Cacabelos, R., editor, *Pharmacoepigenetics*, volume 10 of *Translational Epigenetics*, chapter 34, pages 847–857. Academic Press.

[Barnea et al., 2016] Barnea, E., Mairon, R., and Ben-Shahar, O. (2016). Colour-agnostic shape-based 3D fruit detection for crop harvesting robots. *Biosystems Engineering*, 146:57–70.

[Bartlett et al., 2002] Bartlett, P., Boucherou, S., and Lugosi, G. (2002). Model selection and error estimation. *Machine Learning*, 48(1–3):85–113.

[Bender et al., 2020] Bender, A., Whelan, B., and Sukkarieh, S. (2020). A high-resolution, multimodal data set for agricultural robotics: A Ladybird's-eye view of Brassica. *Journal of Field Robotics*, 37(1):73–96.

[Blok et al., 2016] Blok, P. M., Barth, R., and van den Berg, W. (2016). Machine vision for a selective broccoli harvesting robot. In *5th IFAC AGRICONTROL Conference*, pages 66–71.

[Blok et al., 2020] Blok, P. M., van Evert, F. K., Tielen, A. P. M., van Henten, E. J., and Kootstra, G. (2020). The effect of data augmentation and network simplification on the image-based detection of broccoli heads with Mask R-CNN. *Journal of Field Robotics*, 38(1):85–104.

[Blok et al., 2021] Blok, P. M., van Henten, E. J., van Evert, F. K., and Kootstra, G. (2021). Image-based size estimation of broccoli heads under varying degrees of occlusion. *Biosystems Engineering*, 208:213–233.

[Boser et al., 1992] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152.

[Burges, 1998] Burges, C. J. C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2:121–167.

[Cao et al., 2021] Cao, J., Song, C., Song, S., Xiao, F., Zhang, X., Liu, Z., and Ang, M. H. (2021). Robust Object Tracking Algorithm for Autonomous Vehicles in Complex Scenes. *Remote Sensing*, 13(16).

[Casada et al., 1989] Casada, J. H., Walton, L. R., and Bader, M. J. (1989). Single pass harvesting of broccoli. *Applied Engineering in Agriculture.*

[Castrejón et al., 2017] Castrejón, L., Kundu, K., Urtasun, R., and Fidler, S. (2017). Annotating Object Instances with a Polygon-RNN. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4485–4493.

[Chebrolu et al., 2017] Chebrolu, N., Lottes, P., Schaefer, A., Winterhalter, W., Burgard, W., and Stachniss, C. (2017). Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. *The International Journal of Robotics Research*, 36:1045–1052.

[Chen et al., 2018] Chen, L., Karkee, M., He, L., Wei, Y., and Zhang, Q. (2018). Evaluation of a Leveling System for a Weeding Robot under Field Condition. *IFAC-PapersOnLine*, 51:368–373.

[Choi et al., 2013] Choi, C., Trevor, A. J., and Christensen, H. I. (2013). RGB-D edge detection and edge-based registration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1568–1575.

[Comba et al., 2018] Comba, L., Biglia, A., Ricauda Aimonino, D., and Gay, P. (2018). Unsupervised detection of vineyards by 3D point-cloud UAV photogrammetry for precision agriculture. *Computers and Electronics in Agriculture*, 155:84–95.

[Cousins and Rusu, 2011] Cousins, S. and Rusu, R. B. (2011). 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation*.

[Del Moral, 1996] Del Moral, P. (1996). Non Linear Filtering: Interacting Particle Solution. *Markov Processes and Related Fields*, 2(4):555–580.

[Del Moral and Doucet, 2014] Del Moral, P. and Doucet, A. (2014). Particle methods: An introduction with applications. *ESAIM: Proceedings*, 44:1–46.

[Dobmac Agricultural Machinery, 2021] Dobmac Agricultural Machinery (2021). Dobmac Mechanical Broccoli Harvester. `https://dontstopliving.net/the-worlds-first-broccoli-harvester/`. [Online; accessed: 02/Dec/2021].

[Doucet, 2001] Doucet, A. (2001). *Sequential Monte Carlo Methods in Practice.* Springer, New York, NY.

[Doucet and Johansen, 2009] Doucet, A. and Johansen, A. (2009). A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later. *Handbook of Nonlinear Filtering*, 12.

[Duckett et al., 2018] Duckett, T., Pearson, S., Blackmore, S., Grieve, B., et al. (2018). Agricultural Robotics: The Future of Robotic Agriculture. *UK-RAS Network White Papers. arXiv:1806.06762.*

[Elfring et al., 2021] Elfring, J., Torta, E., and van de Molengraft, R. (2021). Particle filters A hands-on tutorial. *Sensors*, 21(438):1–28.

[Erkan and Dogan, 2019] Erkan, M. and Dogan, A. (2019). Harvesting of Horticultural Commodities. In Yahia, E. M., editor, *Postharvest Technology of Perishable Horticultural Commodities*, chapter 5, pages 129–159. Woodhead Publishing.

[Everingham et al., 2010] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The PASCAL Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, (88):303–338.

[Fahey, 2016] Fahey, J. W. (2016). Brassica: Characteristics and Properties. In Caballero, B., Finglas, P. M., and ToldrÃ¡, F., editors, *Encyclopedia of Food and Health*, pages 469–477. Academic Press, Oxford.

[FANUC UK Limited, 2019] FANUC UK Limited (2019). Automated broccoli harvester turns to FANUC for a helping hand. `https://www.fanuc.eu/uk/en/who-we-are/news/uk-kms-automated-harvester-08-2018`. [Online; accessed: 14/Sep/2019].

[FAO, 2022] FAO (2022). Production quantities of cauliflowers and broccoli by country, 2020. Food and Agriculture Organization of the United Nations http://www.fao.org/faostat/en. Retrieved: 18 January 2020.

[Feng, 2021] Feng, Q. (2021). *End-Effector Technologies*, chapter VII, pages 191–212. Springer International Publishing, Cham.

[Follmann et al., 2019] Follmann, P., König, R., Härtinger, P., Klostermann, M., and Böttger, T. (2019). Learning to See the Invisible: End-to-End Trainable Amodal Instance Segmentation. In *EEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1328–1336.

[Gai et al., 2015] Gai, J., Tang, L., and Steward, B. (2015). Plant recognition through the fusion of 2D and 3D images for robotic weeding. In *2015 ASABE Annual International Meeting*, page 1. American Society of Agricultural and Biological Engineers.

[García-Manso et al., 2021] García-Manso, A., Gallardo-Caballero, R., García-Orellana, C. J., González-Velasco, H. M., and Macías-Macías, M. (2021). Towards selective and automatic harvesting of broccoli for agri-food industry. *Computers and Electronics in Agriculture*, 188:106263.

[Geiger et al., 2012] Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361.

[Gordon et al., 1993] Gordon, N. J., Salmond, D., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140.

[Grilli et al., 2017] Grilli, E., Menna, F., and Remondino, F. (2017). A review of point clouds segmentation and classification algorithms. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W3:339–344.

[Grooms and Robinson, 2021] Grooms, I. and Robinson, G. (2021). A hybrid particle-ensemble Kalman Filter for problems with medium nonlinearity. *PLOS ONE*, 16(3):1–20.

[Guo et al., 2018] Guo, Q., Wu, F., Pang, S., Zhao, X., Chen, L., Liu, J., Xue, B., Xu, G., Li, L., Jing, H., and Chu, C. (2018). Crop 3D-a LiDAR based platform for 3D high-throughput crop phenotyping. *Sci China Life Sci.*, 61(3):328–339.

[Hamuda et al., 2016] Hamuda, E., Glavin, M., and Jones, E. (2016). A survey of image processing techniques for plant extraction and segmentation in the field. *Computers and Electronics in Agriculture*, 125:184–199.

[Hastie et al., 2004] Hastie, T., Rosset, S., Tibshirani, R., and Zhu, J. (2004). The Entire Regularization Path for the Support Vector Machine. *Journal of Machine Learning Research*, 5(1):1391–415.

[He et al., 2017] He, K., Gkioxari, G., DollÃ¡r, P., and Girshick, R. (2017). Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[Holz et al., 2011] Holz, D., Holzer, S., Rusu, R. B., and Behnke, S. (2011). Real-Time Plane Segmentation Using RGB-D Cameras. In Röfer, T., Mayer, N. M., Savage, J., and Saranlı, U., editors, *RoboCup 2011: Robot Soccer World Cup XV*, pages 306–317, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Holzer et al., 2012] Holzer, S., Rusu, R. B., Dixon, M., Gedikli, S., and Navab, N. (2012). Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2684–2689.

[Jiang et al., 2018] Jiang, M., Wu, Y., Zhao, T., Zhao, Z., and Lu, C. (2018). Point-SIFT: A SIFT-like Network Module for 3D Point Cloud Semantic Segmentation.

arXiv e-prints.

[Jimenez et al., 2000] Jimenez, A. R., Ceres, R., and Pons, J. L. (2000). A survey of computer vision methods for locating fruit on trees. *Transactions of the American Society of Agricultural Engineers (ASAE)*, 43(6):1911.

[Jinan and Raveendran, 2016] Jinan, R. and Raveendran, T. (2016). Particle Filters for Multiple Target Tracking. *Procedia Technology*, 24:980–987. International Conference on Emerging Trends in Engineering, Science and Technology (ICETEST - 2015).

[Joshi et al., 2018] Joshi, R. C., Joshi, M., Singh, A. G., and Mathur, S. (2018). Object Detection, Classification and Tracking Methods for Video Surveillance A Review. In *4th International Conference on Computing Communication and Automation (ICCCA)*, pages 1–7.

[Joshi and Mewada, 2020] Joshi, Y. and Mewada, H. (2020). Visual object tracking methodś analysis and impact of deep learning in tracking applications a comprehensive review. *International Journal of Advanced Science and Technology*, 29(3):11931–11945.

[Kamilaris and Prenafeta-Boldú, 2018] Kamilaris, A. and Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147:70–90.

[Karkee et al., 2021] Karkee, M., Bhusal, S., and Zhang, Q. (2021). *Sensors II: 3D Sensing Techniques and Systems*, chapter Chapter 3, pages 39–77. Springer International Publishing, Cham.

[Khanal et al., 2019] Khanal, K., Bhusal, S., Karkee, M., Scharf, P., and Zhang, Q. (2019). Design of improved and semi-automated Red Raspberry Cane Bundling and taping machine based on field evaluation. *Transactions of the ASABE*, 62(3):821–829.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv e-prints*, page arXiv:1412.6980.

[Kootstra et al., 2020] Kootstra, G., Bender, A., Perez, T., and van Henten, E. J. (2020). *Robotics in Agriculture*, pages 1–19. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Kootstra et al., 2021] Kootstra, G., Wang, X., Blok, P. M., Hemming, J., and van Henten, E. (2021). Selective harvesting robotics: current research, trends, and future directions. *Current Robotics Reports*, 2:95–104.

[Krizhevsky et al., 2017] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*, 60(6):84–90.

[Ku et al., 2018] Ku, J., Mozifian, M., Lee, J., Harakeh, A., and Waslander, S. L. (2018). Joint 3D Proposal Generation and Object Detection from View Aggregation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8.

[Kusumam et al., 2016] Kusumam, K., Krajník, T., Pearson, S., Cielniak, G., and Duckett, T. (2016). Can you pick a broccoli? 3D-vision based detection and localisation of broccoli heads in the field. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Korea. IEEE.

[Kusumam et al., 2017] Kusumam, K., Krajník, T., Pearson, S., Duckett, T., and Cielniak, G. (2017). 3D-vision based detection, localization, and sizing of broccoli heads in the field. *Journal of Field Robotics*, 34(8):1505–1518.

[Lan et al., 2020] Lan, J.-H., Chen, S.-W., Lin, C.-H., Shieh, C.-S., Yeh, S.-A., Tsai, I.-H., Liu, C.-H., Tseng, C.-D., Wang, H.-Y., Wu, J.-M., and Lee, T.-F. (2020). Combining the Kalman Filter and Particle Filter in Object Tracking to Avoid Occlusion Problems. In Parinov, I. A., Chang, S.-H., and Long, B. T., editors, *Advanced Materials*, pages 571–586, Cham. Springer International Publishing.

[Le Louedec, 2021] Le Louedec, J. (2021). 3D point cloud annotation tool. `https://justin-lelouedec.medium.com/3d-point-cloud-annotation-tool-9c942afa1a30`. [Online; accessed: 16/Aug/2022].

[Le Louedec et al., 2020] Le Louedec, J., Li, B., and Cielniak, G. (2020). Evaluation of 3D Vision Systems for Detection of Small Objects in Agricultural Environments. In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, pages 682–689. INSTICC, SciTePress.

[LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep Laerning.

Nature, 521:436–444.

[Li, 2017] Li, B. (2017). 3D Fully Convolutional Network for Vehicle Detection in Point Cloud. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1513–1518.

[Li et al., 2018] Li, Y., Bu, R., Sun, M., and Chen, B. (2018). PointCNN: Convolution On X-Transformed Points. *arXiv preprint arXiv:1801.07791*.

[Liakos et al., 2018] Liakos, K. G., Busato, P., Moshou, D., Pearson, S., and Bochtis, D. (2018). Machine Learning in Agriculture: A Review. *SENSORS*, 18(8):1–2.

[Lin et al., 2014] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (2014). Microsoft COCO: Common Objects in Context. *arXiv e-prints*, page arXiv:1405.0312.

[Liu and Chen, 1998] Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo Methods for Dynamic Systems. *Journal of the American Statistical Association*, 93(443):1032–1044.

[Lo et al., 2021] Lo, L.-Y., Yiu, C., Tang, Y., Yang, A.-S., Li, B., and Wen, C.-Y. (2021). Dynamic object tracking on autonomous uav system for surveillance applications. *Sensors*, 21(7888).

[Lorenčík and Zolotová, 2018] Lorenčík, D. and Zolotová, I. (2018). Object recognition in traffic monitoring systems. In *World Symposium on Digital Intelligence for Systems and Machines (DISA)*, pages 277–282.

[Louedec et al., 2020] Louedec, J. L., Montes, H. A., Duckett, T., and Cielniak, G. (2020). Segmentation and detection from organised 3D point clouds A case study in broccoli head detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 285–293.

[Lu and Young, 2020] Lu, Y. and Young, S. (2020). A survey of public datasets for computer vision tasks in precision agriculture. *Computers and Electronics in Agriculture*, 178:105760.

[Lüling et al., 2021] Lüling, N., Reiser, D., Stana, A., and Griepentrog, H. (2021). Using depth information and colour space variations for improving outdoor robustness for instance segmentation of cabbage. In *IEEE International Conference* *on Robotics and Automation (ICRA)*, pages 2331–2336.

[Luo et al., 2021] Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., and Kim, T.-K. (2021). Multiple object tracking: A literature review. *Artificial Intelligence*, 293(2):103448.

[Maggioni et al., 2010] Maggioni, L., von Bothmer, R., Poulsen, G., and Branca, F. (2010). Origin and domestication of cole crops (Brassica oleracea L.): linguistic and literary considerations. *Economic botany*, 64(2):109–123.

[Mandel et al., 2012] Mandel, J., Cobb, L., and Beezley, J. D. (2012). On the convergence of the ensemble Kalman filter. *Applications of Mathematics*, 56:pages533–541.

[Merkle and Reiterer, 2022] Merkle, D. and Reiterer, A. (2022). Overview of 3D point cloud annotation and segmentation techniques for smart city applications. In Erbertseder, T., Chrysoulakis, N., and Zhang, Y., editors, *Remote Sensing Technologies and Applications in Urban Environments VII*, volume 12269, page 1226903. International Society for Optics and Photonics, SPIE.

[Meyer et al., 2003] Meyer, D., Leisch, F., and Hornik, K. (2003). The support vector machine under test. *Neurocomputing*, 55(1):169–186. Support Vector Machines.

[Miao et al., 2021] Miao, T., Wen, W., Li, Y., Wu, S., Zhu, C., and Guo, X. (2021). Label3DMaize: toolkit for 3D point cloud data annotation of maize shoots. *GigaScience*, 10(5).

[Mo et al., 2021] Mo, C., Davidson, J., and Hohimer, C. (2021). *Robotic Manipulation and Optimization for Agricultural and Field Applications*, chapter VII, pages 159–190. Springer International Publishing, Cham.

[Montes and Cielniak, 2022] Montes, H. A. and Cielniak, G. (2022). Multiple broccoli head detection and tracking in 3D point clouds for autonomous harvesting. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence, workshop on AI for Agriculture and Food Systems.*

[Montes et al., 2019] Montes, H. A., Cielniak, G., and Duckett, T. (2019). Model-based 3D point cloud segmentation for automated selective broccoli harvesting. In Althoefer, K., Konstantinova, J., and Zhang, K., editors, *The 20th Towards Autonomous Robotic System (TAROS) Conference*, volume I of *LNAI 11649*, pages 448–459. Springer.

[Montes et al., 2020] Montes, H. A., Le Louedec, J., Cielniak, G., and Duckett, T. (2020). Real-time detection of broccoli crops in 3D point clouds for autonomous robotic harvesting. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10483–10488.

[Nagraj et al., 2020] Nagraj, G. S., Chouksey, A., Jaiswal, S., and Jaiswal, A. K. (2020). Broccoli. In Jaiswal, A. K., editor, *Nutritional Composition and Antioxidant Properties of Fruits and Vegetables*, chapter 1, pages 5–17. Academic Press.

[Nguyen et al., 2016] Nguyen, T. T., Vandevoorde, K., Wouters, N., Kayacan, E., De Baerdemaeker, J. G., and Saeys, W. (2016). Detection of red and bicoloured apples on tree with an RGB-D camera. *Biosystems Engineering*, 146:33–44.

[Oliveira et al., 2021] Oliveira, L. F. P., Moreira, A. P., and Silva, M. F. (2021). Advances in Agriculture Robotics: A State-of-the-Art Review and Challenges Ahead. *Robotics*, 10(2).

[OMahony et al., 2019] OMahony, N., Campbell, S., Carvalho, A., Krpalkova, L., Riordan, D., and Walsh, J. (2019). Point cloud annotation methods for 3D deep learning. In *13th International Conference on Sensing Technology (ICST)*.

[Orzolek et al., 2012] Orzolek, M. D., Lamont Jr., W. J., Kime, L. F., and Harper, J. K. (2012). Broccoli production. In *Agricultural alternatives series*, Agricultural Alternatives series. Penn State Cooperative Extension.

[Pele and Werman, 2010] Pele, O. and Werman, M. (2010). The Quadratic-Chi Histogram Distance Family. In Daniilidis, K., Maragos, P., and Paragios, N., editors, *European Conference on Computer Vision*, pages 749–762, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Polson and Scott, 2011] Polson, N. G. and Scott, S. L. (2011). Data augmentation for support vector machines. *Bayesian Analysis*, 6(1):1 – 23.

[Porikli and Yilmaz, 2012] Porikli, F. and Yilmaz, A. (2012). *Object Detection and Tracking*, pages 3–41. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Potts and Schmischke, 2022] Potts, D. and Schmischke, M. (2022). Learning multivariate functions with low-dimensional structures using polynomial bases.

Journal of Computational and Applied Mathematics, 403. Cited by: 1; All Open Access, Green Open Access.

[Powers, 2011] Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1):37–63.

[Psiroukis et al., 2022] Psiroukis, V., Espejo-Garcia, B., Chitos, A., Dedousis, A., Karantzalos, K., and Fountas, S. (2022). Assessment of Different Object Detectors for the Maturity Level Classification of Broccoli Crops Using UAV Imagery. *Remote Sensing*, 14(3).

[Pulli et al., 2012] Pulli, K., Baksheev, A., Kornyakov, K., and Eruhimov, V. (2012). Realtime Computer Vision with OpenCV. *Queue*, 10(4):40–56.

[Qi et al., 2018] Qi, C. R., Liu, W., Wu, C., Su, H., and Guibas, L. J. (2018). Frustum PointNets for 3D Object Detection from RGB-D Data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 918–927.

[Qi et al., 2017] Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017). PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108.

[Qiu and Shearer, 1992] Qiu, W. and Shearer, S. A. (1992). Maturity assessment of broccoli using the discrete fourier transform. *Transactions of the ASAE*, 35(6):2057–2062.

[Rambach et al., 2018] Rambach, J., Pagani, A., Schneider, M., Artemenko, O., and Stricker, D. (2018). 6DoF Object Tracking based on 3D Scans for Augmented Reality Remote Live Support. *Computers*, 7(1).

[Ramirez, 2006] Ramirez, R. A. (2006). Computer Vision Based Analysis of Broccoli for Application in a Selective Autonomous Harvester. Msc thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

[Ravindran et al., 2021] Ravindran, R., Santora, M. J., and Jamali, M. M. (2021). Multi-Object Detection and Tracking Based on DNN for Autonomous Vehicles: A Review. *IEEE Sensors Journal*, 21(5):5668–5677.

[Remondino and Fraser, 2006] Remondino, F. and Fraser, C. S. (2006). Digital camera calibration methods: Considerations and comparisons. *The International*

Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 36:266–272.

[RoboVeg Ltd, 2021] RoboVeg Ltd (2021). Automated, selective Broccoli and Pointed Headed Cabbage harvesters. `https://www.roboveg.com/harvesters`. [Online; accessed: 09/Dec/2021].

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention – MICCAI 2015*, pages 234–241. Springer International Publishing.

[Rothwell et al., 1993] Rothwell, C. A., Forsyth, D. A., Zisserman, A., and Mundy, J. L. (1993). Extracting projective structure from single perspective views of 3D point sets. In *4th International Conference on Computer Vision*, pages 573–582.

[Rumelhart, 1986] Rumelhart, D. E. (1986). Learning representations by back-propagating errors. *Nature*, 323:533–536.

[Rusu et al., 2009] Rusu, R. B., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (FPFH) for 3D registration. In *IEEE International Conference on Robotics and Automation*, pages 3212–3217.

[Rusu et al., 2010] Rusu, R. B., Bradski, G., Thibaux, R., and Hsu, J. (2010). Fast 3D recognition and pose using the viewpoint feature histogram. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2155–2162.

[Sa et al., 2017] Sa, I., Lehnert, C., English, A., McCool, C., Dayoub, F., Upcroft, B., and Perez, T. (2017). Peduncle detection of sweet pepper for autonomous crop harvesting—Combined color and 3D information. *IEEE Robotics and Automation Letters*, 2(2):765–772.

[Saito and Rehmsmeier, 2015] Saito, T. and Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE*, 10(3):1–21.

[Sanchez, 2003] Sanchez, D. (2003). Advanced support vector machines and kernel methods. *Neurocomputing*, 55:5–20.

[Sanlier and Guler Saban, 2018] Sanlier, N. and Guler Saban, M. (2018). The Benefits of Brassica Vegetables on Human Health. *Journal of Human Health Research.*

[Sarkar and Wolfe, 1985] Sarkar, N. and Wolfe, R. R. (1985). Computer vision based system for quality separation of fresh market tomatoes. *Transactions of the ASAE*, 28(5):1714–1718.

[Sarkar and Raheman, 2021] Sarkar, P. K. and Raheman, H. (2021). A Comprehensive Review of Mechanized Cabbage Harvesting Systems and Its Present Status in India. *Journal of The Institution of Engineers (India): Series A*.

[Shariati et al., 2019] Shariati, H., Moosavi, H., and Danesh, M. (2019). Application of particle filter combined with extended Kalman filter in model identification of an autonomous underwater vehicle based on experimental data. *Applied Ocean Research*, 82:32–40.

[Shearer et al., 1994] Shearer, S. A., Burks, T. F., Jones, P. T., and Qui, W. (1994). One-dimensional image texture analysis for maturity assessment of broccoli. *Paper American Society of Agricultural Engineers*, 1(94-3017).

[Shearer et al., 1991a] Shearer, S. A., Jones, P. T., and Casada, J. H. (1991a). Development of a mechanized selective harvester for cole crops. *Paper American Society of Agricultural Engineers*, 1(91-1018).

[Shearer et al., 1990] Shearer, S. A., Jones, P. T., Casada, J. H., and Swetnam, L. D. (1990). Multiple-pass selective broccoli harvester field trial. *Paper American Society of Agricultural Engineers*, 1(90-1611).

[Shearer et al., 1991b] Shearer, S. A., Jones, P. T., Casada, J. H., and Swetnam, L. D. (1991b). A cut-off saw mechanism for selective harvest of broccoli. *Transactions of the ASAE*, 34(4):1623–1628.

[Shelhamer et al., 2017] Shelhamer, E., Long, J., and Darrell, T. (2017). Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651.

[Silwal et al., 2017] Silwal, A., Davidson, J. R., Karkee, M., Mo, C., Zhang, Q., and Lewis, K. (2017). Design, integration, and field evaluation of a robotic apple harvester. *Journal of Field Robotics*, 34(6):1140–1159.

[Silwal et al., 2016] Silwal, A., Karkee, M., and Zhang, Q. (2016). A hierarchical approach to apple identification for robotic harvesting. *Transactions of the ASABE*, 59(5):1079–1086.

[Silwal et al., 2021] Silwal, A., Prahar, T., and Baweja, H. (2021). *Advanced Learning and Classification Techniques for Agricultural and Field Robotics*, chapter Chapter 13, pages 337–364. Springer International Publishing, Cham.

[Simonyan and Zisserman, 2015] Simonyan, K. and Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556.

[Sites and Delwiche, 1988] Sites, P. W. and Delwiche, M. J. (1988). Computer vision to locate fruit on a tree. *Transactions of the ASAE*, 31(1):257–265.

[Soule and Sides, 1988] Soule, H. M. and Sides, S. E. (1988). Engineering aspects of the Maine broccoli industry. *Paper American Society of Agricultural Engineers*, 1(88-1577).

[Speekenbrink, 2016] Speekenbrink, M. (2016). A tutorial on particle filters. *Journal of Mathematical Psychology*, 73:140–152.

[Szeliski, 2022] Szeliski, R. (2022). *Computer Vision: Algorithms and Applications*. Springer, 2nd edition.

[Tu et al., 2007] Tu, K., Ren, K., Pan, L., and Li, H. (2007). A study of broccoli grading system based on machine vision and neural networks. In *International Conference on Mechatronics and Automation*, pages 2332–2336. IEEE.

[Univerco, 2021] Univerco (2021). Broccoli and cabbage harvester COMMANDER III. `https://univerco.com/en/product/broccoli-and-cabbage-harvester-commander-iii/`. [Online; accessed: 18/Dec/2021].

[Valin et al., 2014] Valin, H., Sands, R. D., van der Mensbrugghe, D., Nelson, G. C., Ahammad, H., Blanc, E., Bodirsky, B., Fujimori, S., Hasegawa, T., Havlik, P., Heyhoe, E., Kyle, P., Mason-D'Croz, D., Paltsev, S., Rolinski, S., Tabeau, A., van Meijl, H., von Lampe, M., and Willenbockel, D. (2014). The future of food demand: understanding differences in global economic models. *Agricultural Economics*, 45(1):51–67.

[van Henten et al., 2009] van Henten, E. J., Van't Slot, D. A., Hol, C. W. J., and Van Willigenburg, L. G. (2009). Optimal manipulator design for a cucumber harvesting robot. *Computers and electronics in agriculture*, 65(2):247–257.

[van Klompenburg et al., 2020] van Klompenburg, T., Kassahun, A., and Catal, C. (2020). Crop yield prediction using machine learning A systematic literature review. *Computers and Electronics in Agriculture*, 177:105709.

[Vrochidou et al., 2022] Vrochidou, E., Nikoleta Tsakalidou, V., Kalathas, I., Gkrimpizis, T., Pachidis, T., and Kaburlasos, V. G. (2022). An Overview of End Effectors in Agricultural Robotic Harvesting Systems. *Agriculture (Basel)*, 12(1240).

[Walton and Casada, 1988] Walton, L. R. and Casada, J. H. (1988). Evaluation of Broccoli Varieties for Mechanical Harvesting. *Applied Engineering in Agriculture*, 4:5–7.

[Wang et al., 2019] Wang, B., Wu, V., Wu, B., and Keutzer, K. (2019). LATTE: Accelerating LiDAR Point Cloud Annotation via Sensor Fusion, One-Click Annotation, and Tracking.

[Wang et al., 2021] Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2021). Scaled-YOLOv4: Scaling Cross Stage Partial Network. In *EEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13024–13033.

[Wang et al., 2017] Wang, X., Li, T., Sun, S., and Corchado, J. M. C. (2017). A Survey of Recent Advances in Particle Filters and Remaining Challenges for Multitarget Tracking. *Sensors*, 17(12).

[Wang and Jia, 2019] Wang, Z. and Jia, K. (2019). Frustum ConvNet: Sliding Frustums to Aggregate Local Point-Wise Features for Amodal 3D Object Detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1742–1749.

[Wilhoit et al., 1990] Wilhoit, J. H., Byler, R. K., Koslav, M. B., and Vaughan, D. H. (1990). Broccoli head sizing using image texture analysis. *Transactions of the ASAE*.

[Wilhoit and Vaughan, 1991] Wilhoit, J. H. and Vaughan, D. H. (1991). A powered cutting device for selectively harvesting broccoli. *Applied Engineering in Agriculture*, 7(1):14–20.

[Wolfe and Swaminathan, 1987] Wolfe, R. R. and Swaminathan, M. (1987). Determining orientation and shape of bell peppers by machine vision. *Transactions of the ASAE*, 30(6):1853–1856.

[Yang and Xu, 2021] Yang, B. and Xu, Y. (2021). Applications of deep-learning approaches in horticultural research: a review. *Horticulture Research*, 8(123):1–31.

[You et al., 2019] You, S., Zhu, H., Li, M., and Li, Y. (2019). A review of visual trackers and analysis of its application to mobile robot.

[Zeng et al., 2017] Zeng, A., Yu, K.-T., Song, S., Suo, D., Walker, J., Rodriguez, A., and Xiao, J. (2017). Multi-view Self-supervised Deep Learning for 6D Pose Estimation in the Amazon Picking Challenge. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1386–1383.

[Zhang et al., 2020] Zhang, D., Chun, J., Cha, S. K., and Kim, Y. (2020). Spatial Semantic Embedding Network: Fast 3D Instance Segmentation with Deep Metric Learning. *ArXiv*, abs/2007.03169.

[Zhang and Karkee, 2021] Zhang, Q. and Karkee, M. (2021). *Agricultural and Field Robotics: An Introduction*, chapter I, pages 1–10. Springer International Publishing.

[Zhang et al., 2021] Zhang, X.-Q., Jiang, R.-H., Fan, Chen-Xiang ans Tong, T.-Y., Wang, T., and Huang, P.-C. (2021). Advances in deep learning methods for visual tracking literature review and fundamentals. *International Journal of Automation and Computing*, 18:311–333.

[Zhao et al., 2016] Zhao, Y., Gong, L., Huang, Y., and Liu, C. (2016). A review of key techniques of vision-based control for harvesting robot. *Computers and Electronics in Agriculture*, 127:311–323.

[Zhao et al., 2019] Zhao, Z.-Q., Zheng, P., Xu, S.-T., and Wu, X. (2019). Object Detection With Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232.

[Zhou et al., 2020] Zhou, C., Hu, J., Xu, Z., Yue, J., Ye, H., and Yang, G. (2020). A monitoring system for the segmentation and grading of broccoli head based on deep learning and neural networks. *Frontiers in Plant Science*, 11:402.

[Zhou et al., 2018] Zhou, Q.-Y., Park, J., and Koltun, V. (2018). Open3D: A Modern Library for 3D Data Processing. *arXiv:1801.09847*.

[Zhu et al., 2018] Zhu, L., Li, Z., Li, C., Wu, J., and Yue, J. (2018). High performance vegetable classification from images based on AlexNet deep learning model. *International Journal of Agricultural and Biological Engineering*, 11(4):217–223.

[Zimmer et al., 2019] Zimmer, W., Rangesh, A., and Trivedi, M. (2019). 3D BAT: A Semi-Automatic, Web-based 3D Annotation Toolbox for Full-Surround, Multi-Modal Data Streams. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1816–1821.