Real-time detection of broccoli crops in 3D point clouds for autonomous robotic harvesting

Hector A. Montes, Justin Le Louedec, Grzegorz Cielniak and Tom Duckett[1]

Abstract— Real-time 3D perception of the environment is crucial for the adoption and deployment of reliable autonomous harvesting robots in agriculture. Using data collected with RGB-D cameras under farm field conditions, we present two methods for processing 3D data that reliably detect mature broccoli heads. The proposed systems are efficient and enable real-time detection on depth data of broccoli crops using the organised structure of the point clouds delivered by a depth sensor. The systems are tested with datasets of two broccoli varieties collected in planted fields from two different countries. Our evaluation shows the new methods outperform state-of-the-art approaches for broccoli detection based on both 2D vision-based segmentation techniques and depth clustering using the Euclidean proximity of neighbouring points. The results show the systems are capable of accurately detecting the 3D locations of broccoli heads relative to the vehicle at high frame rates.

I. INTRODUCTION

In recent years, for a variety of social and economical factors, there has been a widespread interest in the automation of harvesting of fruits and vegetables. One major challenge to the agriculture industry is the access to affordable labour, as it constitutes an important part of its production costs and has become progressively expensive and scarce due to a number of uncertainties ranging from political pressures to migration dynamics [1]. It is therefore desirable to develop autonomous systems to harvest at human-level competence, in terms of both speed and accuracy, to reduce labour and other operational costs [2]. Robots for agricultural tasks have been actively developed for over three decades. However, harvesting is still a labour intensive activity commonly done by hand due to the limited performance of current robotic platforms. Consequently, developing an autonomous robotic harvester capable of reliably identifying vegetable crops under real-time field conditions is essential to increase yield and to better control production costs.

A selective robotic harvester usually consists of three independent systems, namely, 1) a detection system to identify and locate the produce, 2) a picking system to physically manipulate and cut the crop, and 3) a navigation system to allow the robot to move around the field. A key component of a harvesting robot is the recognition system to accurately detect the crop and further determine if it meets the required quality parameters (e.g, size, colour, ripeness), as a precise detection allows the harvesting system to physically collect the crop and then move on to the next harvesting point.

In this paper we focus on the detection system for harvesting crops of broccoli plants and tackle the problem of detecting broccoli heads in real-time.

Real-time 3D perception of the environment remains a major challenge and is crucial for the adoption and deployment of reliable autonomous harvesting robots. This paper describes two detection systems that operate at high frame rates on datasets collected with low-cost RGB-D sensors. The systems' performance has been evaluated using datasets acquired in open field conditions in locations in the UK and Spain. The underlying methods achieve high performance in detection by exploiting the organized structure of the point clouds being processed. The presented methods comprises the following contributions: (1) two efficient systems that run in real-time and process depth information from point clouds at a speed of 10 to 15 fps; (2) a simple and effective algorithm for clustering point clouds that extracts regions based on depth boundaries and normal vector angular properties, (3) an extensive evaluation of the methods on datasets of different broccoli varieties collected from two countries under real-world field conditions. The experimental evaluation shows that our methods outperforms recent state-of-the-art methods based on both 2D vision-based segmentation techniques [3], and 3D clustering based on spatial distances [4]. Our goal is to develop and demonstrate machine learning algorithms capable of delivering high classification performance and real-time execution.

In the next section, we give a brief overview of related work. The methodology followed in this work is presented in Section II, while Section III details our experimental setup and the evaluation results. The paper is then concluded in Section IV.

A. Related work

One major challenge in autonomous robotic harvesting is to reliably identify and locate the crop from other elements in the background at real-time operating speeds. One common approach actively being developed uses 2D images as sensory input, as reflected by the scores of computer vision based techniques available in the literature [5]. However, 3D sensors provide information with sufficient accuracy about the distance from the sensor to the scene, at the expense of some more computing power.

Over the past few years, deep learning techniques have been successfully applied in various fields, and have also recently gained momentum in agriculture [6]. Deep learning techniques have achieved both high classification performance and real-time execution. Nevertheless, these ap-

proaches are data hungry, as the complexity of problems requires large amounts of training data and a considerable annotation effort, both often not readily available and usually expensive and time consuming to acquire for applications in agriculture. Bender *et al.* [7] collected a long-term dataset of broccoli and cauliflower at different growth stages, and demonstrated crop detection using a Convolutional Neural Network (CNN) with a mean average precision (mAP) of 95%. However, this result is for the entire broccoli plant and not the individual broccoli head, which is necessary for autonomous harvesting operations. Other approaches for detection and segmentation based on CNNs for different crops have also been published in recent years [8], [9].

Several studies on the detection of individual broccoli crops can be found in the literature. The very first attempt to accurately locate broccoli heads was developed by Ramirez [10] using a small set of 13 colour images. The method detected the broccoli crop on the entire plant based on contrast and statistical texture analysis. Blok *et al.* [3] described a system for detecting broccoli heads of two different crop varieties based on a Law's texture filter and colour analysis of the head appearance using colour images. They reported a precision score of 99.5%, a recall score of 91.2%, and a negative predictive value of 69.7%. The mean processing time was around 300 ms per image. More recently, Kusumam *et al.* [4] detected broccoli heads in depth images collected with an RGB-D sensor. They combine an Euclidean clustering method, a Viewpoint Feature Histogram (VFH) descriptor, a Support Vector Machine (SVM) classifier, and a temporal filter to detect the broccoli heads. On two broccoli varieties they reported an average precision of 95.2% and 84.5%, respectively, and a mean processing time of 5-6 seconds per depth image.

As these last two results are the best reported in the literature for broccoli detection, we use them in our experimental evaluation as our baseline. We also implement some basic improvements to the system reported in [4] to boost its time execution and to ensure a more extensive evaluation.

II. METHODOLOGY

Affordable depth sensors, like the Intel Realsense suite or the now discontinued Microsoft Kinect, are able to simultaneously capture high-resolution colour and depth images at high frame rates. These depth images can be converted into organized point clouds, *i.e.*, clouds of 3D points that maintain a two-dimensional matrix layout, which are useful in a wide variety of robotics applications. It is also possible to convert stereo images into organized point clouds provided that the camera's intrinsic calibration parameters are known. The main advantage of these organized point clouds is that the location of neighbouring points of any other point within the matrix grid can be retrieved in constant time, making it unnecessary to run costly searches and drastically speeding up processing times. In our study we extensively use this property to process depth data at high frame rates.

To ensure a more complete and fair evaluation of our system, we have implemented a slightly modified yet faster

version of the method presented by Kusumam *et al.* in [4], here dubbed Fast Euclidean Clustering (FEC), and its results are compared against our two real-time 3D perception systems called Organised Edges Segmentation (OES) and Organised Region Growing Segmentation (ORG).

The detection pipeline from [4] included a statistical outlier removal step, depth-range filtering, Euclidean cluster extraction, normal estimation, feature extraction, classification, and an additional temporal filtering step to improve the overall classification results. In our FEC pipeline, the outlier removal and the temporal filter steps were removed, as the former is computationally expensive and the filtered points do not affect the clustering step, and the latter is a procedure added to further improve the classification performance. Nevertheless, the highest score is used in our comparative evaluation. We also replaced the the normal estimation step by the faster Integral Images Normals Estimation method briefly described in Section II-B.1.

A. Organised clustering pipeline

The two systems presented in this paper involve four main steps: (1) a preprocessing stage for normal estimation and for detecting edges on organized point clouds (OES only) that uses the depth information of neighbouring points to estimate edge boundaries, (2) a clustering step which returns a list of clusters of points with common attributes, (3) a feature extraction step to encode cluster properties, and (4) a classifier for model learning and to predict broccoli detections, as shown in Figure 1. We use the algorithms available as part of the PCL library [11] for processing point clouds.

B. Preprocessing

1) Integral images normal estimation: Normal estimation methods commonly work by either computing the normal 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. Searching in a neighbourhood is commonly a slow process for many practical applications in robotics. Integral images provides a method for normal estimation on organized clouds [12]. The algorithm uses the cloud as a depth image while keeping a 2D *matrix* layout of the data. This allows to quickly create rectangular areas over which the normals are computed by taking advantage of the relationship between neighbouring points without the need for costly searches. The result is a very efficient method to compute normal vectors using the inherent grid structure of the point clouds collected by low-cost RGB-D sensors. We use this algorithm to estimate normals for the FEC, OES and ORG methods.

2) Organized 3D edge detection: The 3D edge detection algorithm, introduced by Choi *et al.* [13], labels points as edges based on point depth discontinuities. 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 organized layout of the point cloud. Similarly, the algorithm uses a
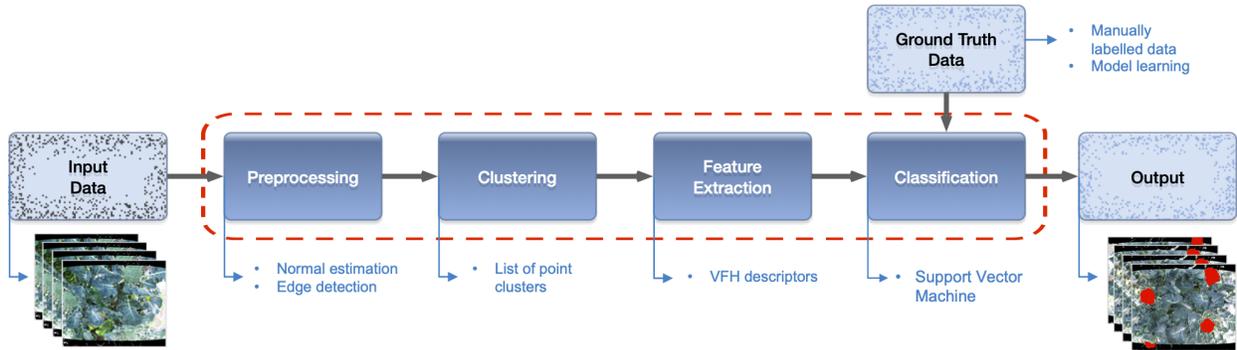
Fig. 1. Real-time broccoli detection system pipeline. The main steps of the pipeline are highlighted by a red bounding box. The input data are the point cloud frames acquired by the RGB-D sensor. The output is the locations of the broccoli heads detected on each input frame.

neighbourhood size to label points as one of the predefined types of edges (*occluding* and *occluded* in this paper). The method outputs a set of edge labels that corresponds to all points in the point cloud.

C. Clustering

The output of this step is a list of clusters. Each cluster groups points together either because they belong to the same area bounded by an edge (OES), or because the angle of their normal vectors and their surface smoothness are within a similarity threshold (ORG). 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 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. The steps of the clustering process are listed in Algorithm 1.

1) Organized edge clustering: 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 above.

2) Organized region growing clustering: 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 normal curvature (*i.e.*, the amount of change in direction of the surface normal) is also within a threshold value. The idea is to make the region spread to other points in the vicinity with similar surface curvature.

D. Feature extraction

We use the Viewpoint Feature Histogram (VFH) descriptor, a global 3D feature descriptor that uses the normal vector angles to represent the properties of data points within the same cluster. The VFH is computed by estimating a normalized direction vector between the viewpoint (*i.e.*, the

---

Algorithm 1: Organized clustering algorithm

---

1   function OrgClustering($PointCloud, Similaty$);
    Input : A $PointCloud$ and a $Similarity$ function
    Output: A set of $Clusters$
2   Clusters=[] # list of clusters
3   processed=[PointCloud.size, *false*] # points processed
4   **for** *point_ic* **in** *PointCloud* **do**
5   **if** *processed[point_ic]* **then**
6   **loop** # point already processed
7   CurrentCluster=[]
8   processed[point_ic]=*true*
9   **for** *point_cs* **in** *CurrentCluster* **do**
10   # left, right, top & bottom neighboring points
11   n1, n2, n3, n4 = get_four_neighbors(point_cs)
12   **if** ***not*** *processed[n1,n2,n3,n4]* **then**
13   **if** $Similarity(point\_cs, [n1,n2,n3,n4])$
        then
14   **add** PointCloud[n1,n2,n3,n4] **to** CurrentCluster
15   processed[n1,n2,n3,n4]=*true*
16   **if** *max_size* $\geq$ *CurrentCluster* $\geq$ *min_size* **then**
17   **add** CurrentCluster **to** Clusters
18 return Clusters

---

position of the sensor) and the cluster centroid. Then the angles between this vector and the surface normals of all the points in the cluster are computed to form the histogram bins of the descriptor. To make the VFH descriptor invariant to scale, the bins can be normalized using the number of points in each cluster. The VFH descriptor transforms individual point characteristics into cluster features useful to distinguish one cluster from another. For extended details and a discussion on the VFH descriptor properties, the interested reader is referred to [14].

E. Classification

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. All classification results were performed using a Support Vector Machine (SVM) classifier with the following parameters: RBF (radial basis function) kernel, $C = 2.0$, $\gamma = 0.0078125$, and class weights w adjusted inversely proportional to class frequencies in the input data as $w_j = n/(k \times n_j)$, where n is the total number of observations; n_j is the number of observations in class j, and k is the number of classes. These parameters were determined by *k-fold* cross validation ($k = 5$) based on a grid search. The final classification output of the system pipeline is a set of clusters representing the locations of broccoli heads. For experiments on the same 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.

III. Experimental evaluation

A. Datasets

In our experimental evaluation we use two of the same datasets used in the experiments reported by Kusumam *et al.* [4][1]. These datasets were acquired in planted farm fields under different weather conditions using the Kinect 2 sensor (1920×1080 RGB, and 512×424 depth resolution). The sensor was fixed inside an enclosed box mounted on a farm tractor for uniform illumination and for protection from external conditions. The UK set consisted of 600 frames of the broccoli variety Ironman, while the Spain 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 and the enclosure setup are shown in Figure 2.

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. These differences produce points captured at different distances and also make occlusions more evident. 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. In consequence, the average number of clusters extracted per frame 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 these different schemes to evaluate the generalization performance of our systems, and also to help to decide hardware configurations for the robotic harvester. To evaluate our systems, we collected ground truth information by manually annotating all 3D datasets using a software tool built specially for this task.

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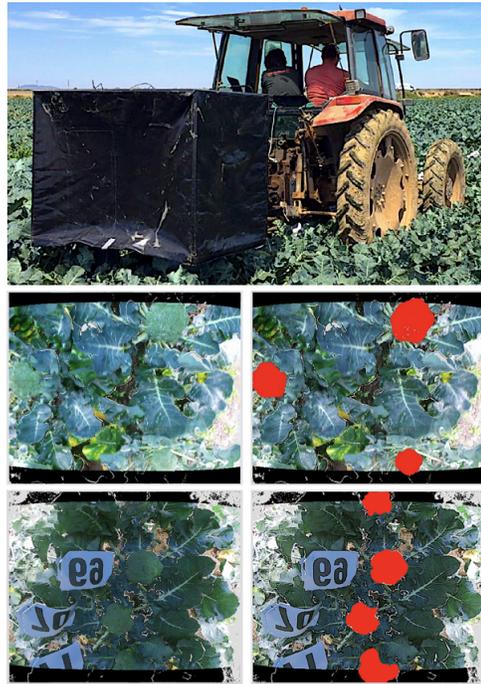

Fig. 2. Shown at the top is the enclosing box mounted at the back of a tractor while collecting data on a broccoli field in Spain. Below are frame samples of the UK dataset (first row) and the Spain dataset (bottom row) collected with this hardware platform. The images on the right column show in red the ground truth annotation 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.

B. Evaluation metrics

1) Confusion matrix evaluation: The purpose of this evaluation is to compare our results with the segmentation pipeline published by Blok *et al.* [3] based on conventional computer vision techniques. Blok's results were evaluated using the ground truth data of 200 images of two broccoli varieties, acquired with a RGB camera (2448×2050 resolution) in fields in The Netherlands. Table I summarizes a comparative list of evaluation results between Blok *et al.* and our detection pipelines. Even though the results of Table I show a high precision score for every detection outcome, conclusions drawn when comparing scores such as accuracy, recall or specificity might be misleading. Accuracy, for instance, is a poor performance indicator for datasets with a highly imbalanced class distribution. For this reason, we also chose to evaluate performance through Cohen's kappa score κ [15], a score that expresses the level of agreement between the classifier predictions and the ground truth data. The scores in Table I for each dataset combination indicates an improved performance from previously published results [3]. In these experiments, we achieve a high agreement to the ground truth in all datasets combinations as the κ score increased from 76.8% to 83.5–96.9%. We also evaluated the negative predictive value (NPV), *i.e.*, the proportion of negative results correctly predicted, which increased from 69.7% to 98.5-99.8%.

	Accuracy	Precision	Recall	Specificity	NPV	κ
Blok *et al.* [3]	92.4%	99.5%	91.2%	97.9%	**69.7%**	**76.8%**
FEC UK v UK	99.4%	99.0%	94.0%	99.9%	99.5%	96.1%
FEC UK v Spain	98.7%	99.7%	85.0%	99.9%	98.6%	91.1%
FEC Spain v Spain	98.8%	96.6%	84.3%	99.8%	99.0%	89.4%
FEC Spain v UK	98.0%	89.6%	78.6%	99.4%	98.5%	82.7%
OES UK v UK	99.9%	98.4%	95.7%	99.9%	**99.8%**	**96.9%**
OES UK v Spain	99.3%	98.3%	86.8%	99.9%	**99.3%**	**91.8%**
OES Spain v Spain	99.2%	96.9%	88.0%	99.8%	**99.3%**	**91.8%**
OES Spain v UK	98.4%	93.0%	77.1%	99.7%	**98.6%**	**83.5%**
ORG UK v UK	98.8%	98.8%	87.7%	99.9%	98.8%	92.3%
ORG UK v Spain	96.6%	98.9%	66.3%	99.8%	96.5%	77.6%
ORG Spain v Spain	98.1%	98.6%	83.0%	99.9%	98.1%	89.1%
ORG Spain v UK	98.0%	96.4%	79.8%	99.7%	98.2%	86.3%

	FEC			OES			ORG		
	UK1	UK2	Spain	UK1	UK2	Spain	UK1	UK2	Spain
UK1	0.98	0.94	0.78	0.96	0.92	0.79	0.99	0.98	0.95
UK2	0.93	0.94	0.84	0.94	0.93	0.84	0.99	0.99	0.97
Spain	0.95	0.94	0.91	0.88	0.86	0.90	0.98	0.97	0.98
UK1	0.96	0.92	0.56	0.86	0.85	0.54	0.70	0.62	0.57
UK2	0.92	0.92	0.61	0.85	0.85	0.58	0.70	0.62	0.59
Spain	0.94	0.93	0.65	0.78	0.80	0.59	0.69	0.60	0.61

2) PRC and IoU overlap evaluation: The second evaluation metric used was a *precision-recall curve* (PRC). 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 very imbalanced [16]. Because a correct detection means that our systems 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 outcome. This overlap was calculated via the Intersection Over Union (IoU) metric. For each IoU threshold, we calculate the PRC of all predicted results. Figure 3 shows selected examples of the average precision score (APS) computed at different discrimination threshold settings of the OES, ORG and FEC detection systems for different IoU threshold values.

The precision is the percentage of correct detections (*i.e.*, a sample predicted as broccoli head has indeed been labelled as *broccoli*), whilst the recall measures how well a sample predicted as leaf, soil or any other background element has been labelled as such. Every PRC plot in Figure 3 shows a higher area under the curve for smaller IoU threshold values. This is expected, but the precision and recall do not express how much of the broccoli head was precisely detected by the different algorithms. To investigate this further, we use the mean average precision (mAP) calculated by averaging the APS over multiple IoU values within the range 0.5-0.95 in 0.05 steps. This detection evaluation complements the traditional average precision computed at a single IoU of 0.5. A complete set of APS values for various datasets combinations between the three broccoli detection pipelines is summarized in Table II.

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 closer together in Fig. 3. 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. Table II shows a reduction in precision as the IoU threshold increases, especially for the OES and the ORG algorithms. These results may suggest some guidelines for the hardware configuration of an autonomous robotic broccoli harvester. Figure 4 shows a set of selected examples for all three detection algorithms.

3) Time performance: One relevant feature of the FEC, ORG and OES systems is that they achieve processing frame rates of 3.2, 9.2 and 14.8 fps, respectively, on an Intel i7 processor at 3.7 GHz clock speed. This processing time is also a significant improvement on related research [3], [4] and is the result of how the 3D space is explored by the algorithms. While the FEC searches for neighbouring points in Euclidean space, both the OES and the ORG methods can retrieve any point information in constant time by exploiting the ordered grid of the point cloud. Because other systems in the robotic harvester also require time to perform their own operations, an efficient detection 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.

IV. CONCLUSIONS

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. Comparative experimental results also show that our methods achieved both high classification performance and real-time execution against the best approaches for broccoli detection based on the Euclidean proximity of points [4] when tested on the same datasets. Our method also performed favourably against another broccoli detection algorithm based on a machine vi-
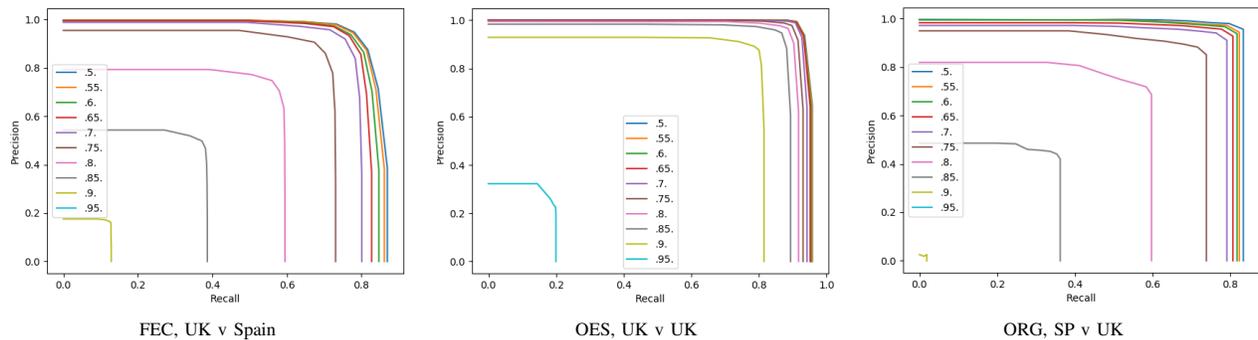
Fig. 3. Three selected PRC plots showing the classification performance of the detection systems on different training and testing datasets combinations. Each curve is the PRC at various discrimination threshold settings for different IoU values within the range 0.5-0.95.
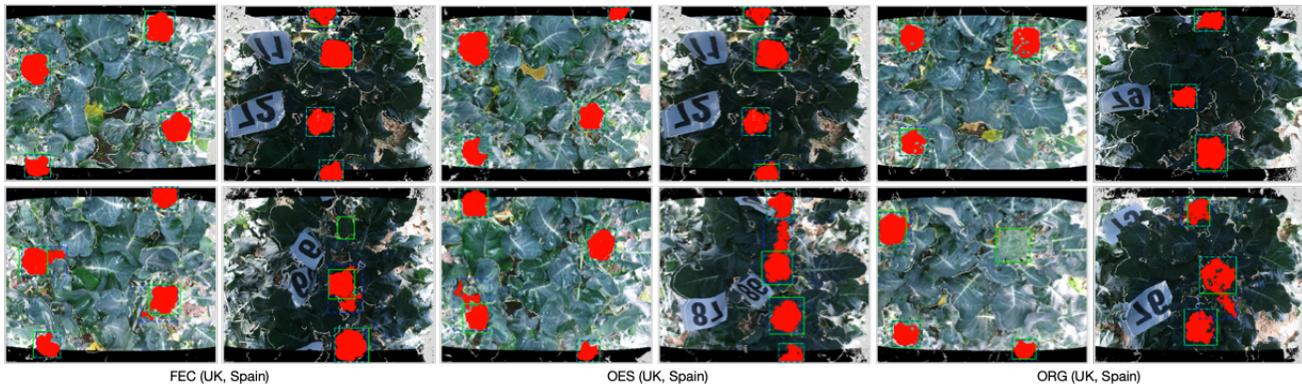

Fig. 4. On the top row are selected examples of some of the best detection results from the three algorithms for both datasets. The bottom row shows a selection of some of the worst detections. The red areas show the detected locations while the green boxes show the true locations.

sion system that uses texture and colour image segmentation [3]. The clustering strategy implemented by both algorithms yields 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 generalization of the proposed systems. An interesting research direction 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. Also, we are currently investigating Convolutional Neural Networks and related deep learning techniques, as they have become the method of choice for many detection and classification problems.

REFERENCES

[1] T. Duckett *et al.*, "Agricultural robotics: The future of robotic rgriculture," *UK-RAS Network White Papers. arXiv:1806.06762*, 2018.
[2] C. W. Bac, E. J. van Henten, J. Hemming, and Y. Edan, "Harvesting robots for high-value crops: State-of-the-art review and challenges ahead," *Journal of Field Robotics*, vol. 31, no. 6, pp. 888–911, 2014.
[3] P. M. Blok *et al.*, "Machine vision for a selective broccoli harvesting robot," in *5th IFAC AGRICONTROL Conference*, 2016, pp. 66–71.
[4] K. Kusumam, T. Krajník, *et al.*, "3D-vision based detection, localization, and sizing of broccoli heads in the field," *Journal of Field Robotics*, vol. 34, no. 8, pp. 1505–1518, 2017.
[5] Y. Zhao, L. Gong, Y. Huang, and C. Liu, "A review of key techniques of vision-based control for harvesting robot," *Computers and Electronics in Agriculture*, vol. 127, pp. 311–323, 2016.
[6] A. Kamilaris *et al.*, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.
[7] A. Bender, B. Whelan, and S. Sukkarieh, "A high-resolution, multimodal data set for agricultural robotics: A Ladybird's-eye view of Brassica," *Journal of Field Robotics*, vol. 37, no. 1, pp. 73–96, 2019.
[8] A. Milioto, P. Lottes, and C. Stachniss, "Real-time semantic segmentation of crop and weed for precision agriculture," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 2229–2235.
[9] A. Wang *et al.*, "Semantic segmentation of crop and weed using an encoder-decoder network and image enhancement method under uncontrolled outdoor illumination," *IEEE Access*, vol. 8, pp. 81 724–81 734, 2020.
[10] R. A. Ramirez, "Computer vision based analysis of broccoli for application in a selective autonomous harvester," MSc thesis, Virginia Polytechnic Institute and State University, 2006.
[11] S. Cousins and R. B. Rusu, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation*, 2011.
[12] S. Holzer, R. B. Rusu, *et al.*, "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*, 2012, pp. 2684–2689.
[13] C. Choi, A. J. Trevor, and H. I. Christensen, "RGB-D edge detection and edge-based registration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1568–1575.
[14] R. B. Rusu *et al.*, "Fast 3D recognition and pose using the viewpoint feature histogram," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 2155–2162.
[15] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.
[16] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, pp. 1–21, Mar. 2015.