



Exploring A Novel Automated Cucumber Harvesting Framework

Shahryar Ghorbanirezaei ¹  and Jill Urbanic² 

¹University of Windsor, ghorbans@uwindsor.ca

²University of Windsor, jurbanic@uwindsor.ca

Corresponding author: Jill Urbanic, jurbanic@uwindsor.ca

Abstract. Cucumbers represent a substantial portion of greenhouse vegetables cultivated in southern Ontario, Canada. The challenges associated with manual labor—such as repetitive motions and limited labor availability—have sparked interest in automation as a potential solution for cucumber harvesting. Image processing plays a critical role in the development of automated systems. This study introduces a novel cucumber detection method that integrates XYZ point cloud and RGB data, utilizing an Intel RealSense D435if 3D camera, and tests a comprehensive semi-automated harvesting framework. The camera's raw output generates a file containing thousands of lines of RGB-XYZ data for each pixel, comprising numerical values for R (red), G (green), B (blue), and X, Y, Z (3D spatial coordinates). By directly employing these RGB-XYZ values, this study identifies cucumber pixels and measures their distance from the camera. The proposed methodology isolates cucumber pixels within the RGB data and determines stem-end positions using XYZ data, optimizing the harvesting process based on the minimum Z-distance. Alongside this approach, bounding box and key-point detection methods based on YOLO (You Only Look Once) are also explored. Experiments conducted in an artificial greenhouse demonstrate the feasibility of the system, achieving successful robotic harvesting when stem-end coordinates are provided semi-automatically. Therefore, future work should existing image detection techniques with the 3D camera analyses to facilitate harvesting and inspection activities within the same system.

Keywords: Image Detection; YOLO; Shape and Color Features; Cucumber Characteristics; Geometry Analysis; 3D Camera.

DOI: <https://doi.org/10.14733/cadaps.2026.725-743>

1 INTRODUCTION

Windsor-Essex in Ontario, Canada has one of the highest densities of greenhouses globally (second to the Netherlands) and most of Canada's greenhouses are located in Ontario, Canada. Statistics prepared by the Canadian Greenhouse Vegetable industry illustrate the greenhouse distribution and the volume of production associated with greenhouse produce [2, 16], and it can be seen that cucumbers production volumes are very high (Fig. 1). Despite housing only 36% of the total number of greenhouses, Ontario greenhouses produce 71% of the greenhouse products, underscoring the

presence of larger greenhouses, more fertile soil and ready availability of water resources in the province. In fact, the distribution of costs indicates that gross yearly payroll is the major expense (29%), operating expenses are 22%, other crop expenses such as fertilizer and pesticides are 19%, and plant material purchases are 15%. Manual cucumber harvesting is common in the greenhouse environments, but this is labor-intensive and the quality, speed, and efficiency of the harvesting heavily relies on the workers' skills, and physical strength. In 2021, approximately 5,000 workers are employed in Canadian cucumber greenhouses, encompassing both permanent and seasonal labor. Particularly during the COVID-19 pandemic, labor shortages posed challenges for greenhouse operations.

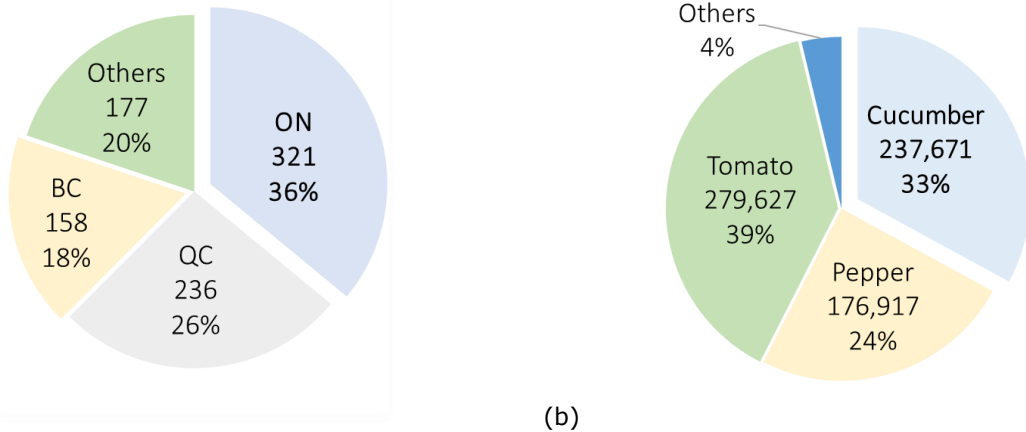


Figure 1: (a) Statistics on the distribution of commercial greenhouses in Canada out of a total of 892 greenhouses, (b) Statistics on the distribution of greenhouse production volumes (in metric tons) in Canada

Manual cucumber harvesting is common in that area, but it is labor-intensive, inconsistent, and physically taxing. Automated methods of harvesting are being considered as a viable alternative, but this is not a mature solution. Research in automated agricultural robots often focuses on one or two specific system components: produce detection, innovative grippers, robotic arm efficiency, autonomous functions, and mechanical harvesting alternatives. However, comprehensive research on semi or fully automated harvesters is limited, with a notable example being the root trimmer developed by McGuinness et al. [13]. This machine can perform lifting, soil removal, root trimming, and sorting of various materials. However, it is mostly a mechanical device; the lack of detection system is evident. Another significant advancement is the dual-arm robot introduced by Yoshida et al. equipped with a Red-Green-Blue (RGB) RGB-D camera and two robotic arms with end effectors [21]. However, their system needs dataset and it can only work for fruit / produce similar to the dataset. Davidson et al. introduced a new apple harvesting gripper, which includes detection [3]. However, modifications like workspace expansion and vehicle mounting are needed for real orchard testing.

There are more studies focusing on detection aspects solely. Image detection is challenging due to the location, orientation, and variable shape characteristics of the produce, as well as the shadows cast due to the leaf and light source locations. Image processing for detection requires specialized platforms, and ImageJ and Fiji, open-source programs, were explored for suitability. In this study, segmentation and thresholding are performed initially, followed by the utilization of a detection plugin to identify the borders and edges of plant components. The results of applying color thresholds and some other filters, are shown in Fig. 2.



Figure 2: (a - d) Image J results for filtering and extracting the boundaries.

The system removed the background and focused on green colors. By utilizing the color deconvolution option, the quality of the image processing is enhanced. Additionally, the edges are eroded to enhance the quality of connecting points and edges as shown in Fig. 3. Unresolved is the ability to accurately extract the boundaries of cucumbers while excluding the vines and leaves' boundaries.



Figure 3: (a - b) The deconvolution option and finding the edges of plant components and final polish on edge detection using the eroded option.

The lack of implementing a robust and extendable detection system in a harvester is evident in much of the reported literature. The implementation strategies are mostly dependent on the dataset and the neural network architecture. Jing et al. developed an apple detection model using a dataset of 1,036 images [9]. Their algorithm combines RGB color data with contour detection to enhance shape recognition and reduce reliance on RGB alone. However, its applicability was restricted to the dataset used. Zheng et al. used shape and RGB features to recognize cucumbers, eggplants, tomatoes, and peppers from a 1,600-sample dataset [20]. Using a RealSense D415 camera, they captured aligned color images and depth maps, applied key point for size estimation, and used a zoom-in technique for better accuracy. While they measured length and diameter in 3D, their size estimation accuracy was limited. Gené-Mola et al. achieved detecting Fuji apples using RGB, depth, and backscatter data with a Microsoft Kinect sensor on a dataset of 967 samples [6]. Their method involved creating a 5-channel RGB-DS image and applying the Faster R-CNN (Region-based Convolutional Neural Network) for region classification. However, the model depended heavily on color data and focused solely on detection, excluding harvesting considerations. Kheiralipour et al. analyzed cucumber shapes using PC1438 and Canon cameras, focusing on diameter and introducing unique shape features like centroid non-homogeneity [10]. After blue channel extraction and image adjustments, they

classified shapes with an artificial neural network. Although their geometry analysis was thorough, the study was limited to specific cucumbers and did not address automated harvesting. Rabby et al. developed a shape and RGB-based detection model for apples and oranges using a modified Canny edge detection algorithm [11]. Their multi-step process included gradient computation, thresholding, edge connection, and weak edge removal, along with segmentation, feature extraction, and training. By the way, details on sample size were not provided, and they did not explore automated harvesting. Liu et al. achieved detecting apples using RGB and shape features on a dataset of 1,844 images, captured with a Canon IXUS 275HS camera [12]. Their method involved Simple Linear Iterative Clustering (SLIC) segmentation, color vector extraction, and Histogram of Oriented Gradients (HOG) vector classification using Support Vector Machine (SVMs). Although effective for detection, their study did not cover automated harvesting. Tu et al. counted passion fruits using RGB and depth features from a dataset of 2,275 samples, acquired with a Microsoft Kinect camera [17]. Their method involved RGB data augmentation and detection followed by fusing results for RGB-D (depth) detection. However, their study lacked a comprehensive device setup. Iskandar et al. Melinjo fruit using RGB and Hue, Saturation, and Value (HSV) features from a dataset of 1,003 samples [7]. Their methodology involved Google Collaboratory for manipulating RGB and HSV data, including masking for background removal and classifying fruits as ripe or raw based on HSV values. However, they did not address the harvesting process, and their model's reliance on a specific dataset limits its applicability to other produce. Eizentals et al. researched pose estimation of peppers using a laser range finder, tested it in a real farm [4]. They partially analyzed geometric aspects of the peppers, but the maximum accuracy of 77.6% is insufficient for practical application in real-world scenarios. A yield real time monitoring system was developed by Chang et al. that is able to map the yield of wild blueberries [1]. Blueberry precise geographic locations are found. They focus on large areas of blueberries and not the exact position for harvesting though.

Based on theoretical aspects, Shaprov et al. explored elements of automated harvesting by modeling the system and determining optimal forces to prevent crop damage [15]. However, their research was limited to theoretical considerations without practical implementation. Manipulation of RGB and *XYZ* point cloud data, and transforming the RGB data to the International Commission on Illumination (CIE) *XYZ* color space is done by Molada-Tebar et al. [14]. They exploited a Python toolbox that improves colorimetric precision in RGB-to-*XYZ* conversions. Using RGB-*XYZ* heightmaps and combined 2D and 3D CNNs, a new method that achieves an 82.4% success rate in grasping diverse objects is proposed by Wang et al. [19]. This includes a two-stage robotic grasp detection. The literature review for these studies is summarized in Fig. 4. It can be seen that solutions for detecting cucumbers, solutions that consider both color and shape, and provide a basis for robot trajectories necessary for harvesting, are limited.

From the literature related to image detection and analysis tools, three of the popular pretrained models are Single Shot Multibox Detector (SSD), Region-based Convolutional Neural Network (R-CNN), and YOLO (Tab. 1). The YOLO family is chosen for this research.

In this study, a new method of detection is put forward and is tested in a robotic harvesting framework. It is hypothesized that manipulating *XYZ* point cloud data simultaneously with the RGB data will provide a basis for effective real-time image processing and harvesting [5]. Cucumber detection using You Only Look Once (YOLO), a common method, is performed, followed by an analysis of our detection approach and the decision-making process. The performance of the cucumber harvesting framework is evaluated to determine whether the approaches are effective. An in-house 'artificial' greenhouse test bed is developed featuring 3D printed peppers, cucumbers, and apples, with some artificial leaves and vines to introduce shadows and interference conditions. Complementary research is performed to detect the cucumber boundaries.

The final proposed system employs a combination of contemporary image detection solutions for boundary detection and *XYZ* & RGB data manipulation for trajectory planning and future quality classification. This is future work.

Pretrained model	Characteristics	Advantages
SSD	Multiscale feature extraction, narrow down target in each stage	1. Balance between speed and accuracy 2. Handling different object sizes
Faster R-CNN	Two-stage approach: 1. RPN, 2. Final object detection	1. High accuracy 2. Better handling of complex object detection
YOLO	Single-pass approach	1. Real-time detection 2. Fast response 3. Easy to implement 4. Diverse range of objects

Table 1: Comparison between three popular pretrained modeling strategies.















































































	Cucumber  Others  None 	RGB  Shape  RGB-XYZ  None 	No need to dataset	Image processing  Decision-making  Robot trajectory/harvesting 	Expandable
McGuinness (2023)			X		X
Yoshida (2022)			X	  	X
Davidson (2016)		 	X	  	X
Jing (2021)		 	X		X
Zheng (2022)	 	 	X		✓
Gené-Mola (2019)		 	X		X
Kheiralipour (2017)		 	X		X
Rabby (2018)		 	X		✓
Liu (2019)		 	X		X
Tu (2020)		 	X		X
Iskandar (2022)			X		X
Eizentals (2016)		 	X		X
Chang (2012)			X		X
Shaprov (2022)			✓		X
Molada-Tebar (2024)			✓		X
Wang (2020)			X	 	X
This study			✓	  	✓

Figure 4: Literature review summary matrix.

2 ROBOTIC CUCUMBER HARVESTING FRAMEWORK

An overview of the complete cucumber harvester is shown in Fig. 5, which includes four primary components: 1. an image processing unit, 2. a robotic arm, 3. a cutter, and 4. a container to hold the cut cucumbers. The focus of this paper is on a new detection method, although the research included all elements. For the first step, the cucumber is detected. Then the position of the cucumber stem end is used as input to generate a robot trajectory (a Universal Robots collaborative robot UR3e [18]). For the third stage, the robotic arm cuts the cucumber from the stem end. In this study, cutting blades were placed in 3D printed brackets, which were embedded in the TCP of the UR3e robot. Finally, the cucumber is placed into a container.

A detailed algorithm of the processes is shown in Fig. 6. First, in the detection phase, the values of the cucumber(s) are kept and all other RGBs like vines and leaves are removed. The positional data for the cucumbers' pixels are also kept. In the second phase is the decision-making where the pixels are sorted based on their distance from the robotic harvester. The pixel with minimum distance in Z is chosen as the cutting point - the robotic arm starts scanning from top (note, repositioning and scanning is repeated from different angles to ensure cucumber detection).

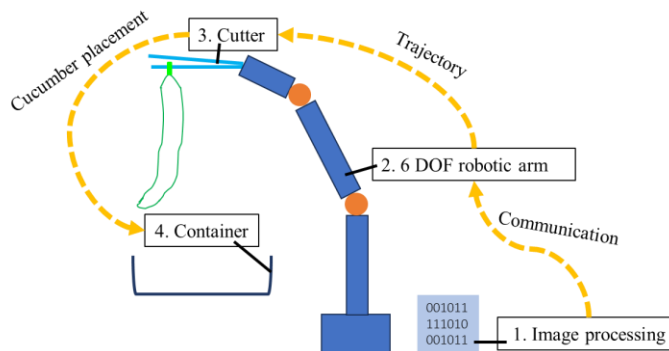


Figure 5: A full automated cucumber harvester includes four main components: a detection system, a robotic arm, a cutting edge, and cucumber placement.

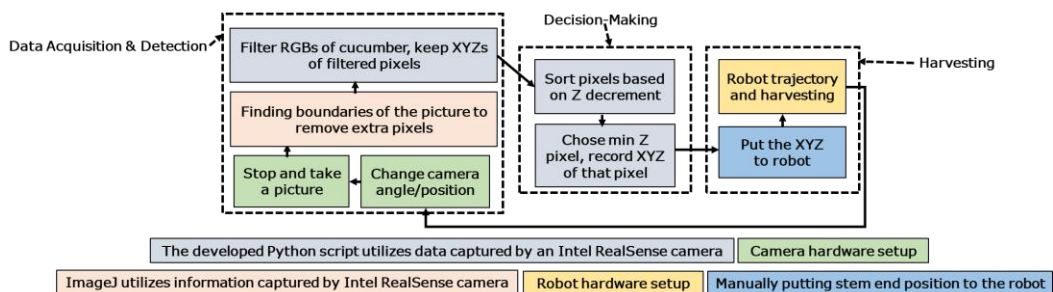


Figure 6: Algorithm for the processes of this study from detection to harvesting.

2.1 Methodology and Test Setup

A compact and lightweight Universal Robots collaborative robotic (cobot) arm (model UR3e) is used as the base test system [18]. An artificial greenhouse featuring 3D printed peppers, cucumbers, and apples, with some artificial leaves and vines, is utilized for the testing. This system has produce of different colors (Fig. 7). The leaves, vines, and cucumbers are very similar in coloration, increasing the detection challenges. As well, the artificial cucumbers can be hidden behind vines and leaves.

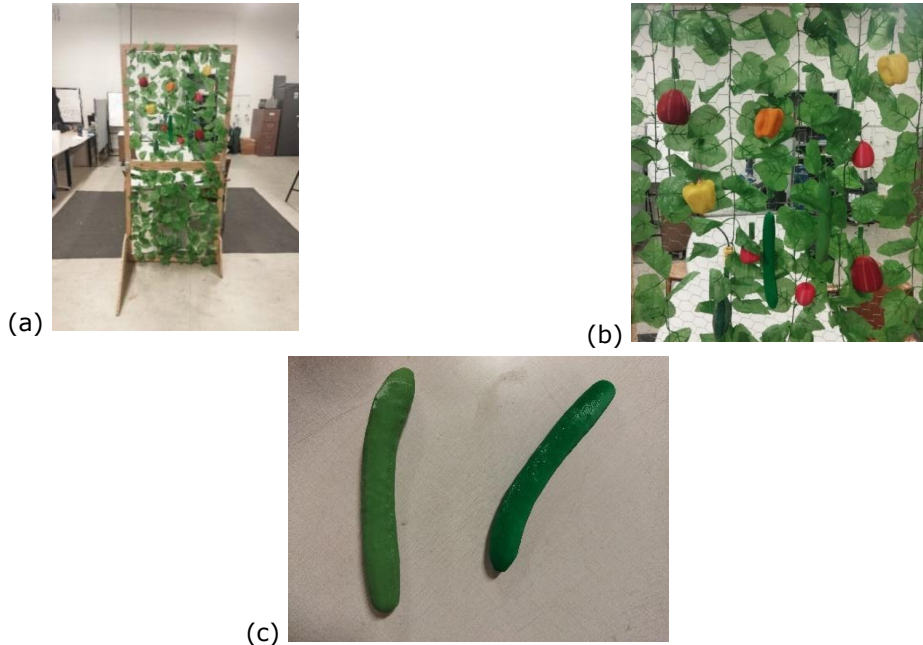


Figure 7: (a) and (b) In-house testing system, (c) fabricated cucumber models

A 3D camera is employed to collect the color and shape data, and for this research, an Intel RealSense D435if camera is utilized. Intel's RealSense D435if camera employs LiDAR and stereo depth technologies [19]. Operating on USB 5V power, the D435if captures color and XYZ data simultaneously, producing a point cloud for scene reconstruction. Widely utilized in machine-vision-based devices due to its compact design, this 3D camera proves particularly useful when the focus is on measuring the distance between the object and the camera. The structure of the output of Intel's RealSense D435if 3D camera is:

```

Pixel 1
RGB (first line)
XYZ (second line)
...
...
Pixel 50000
RGB (first line)
XYZ (second line)

```

To mount the camera on the UR3e arm, a bracket is 3D printed using a Creality-Halton-Mega Pro (a resin-based system). In this study, the motion of camera and cutter is controlled by graphical programming interfaces and teach pendants of UR3e. If the camera moves too quickly, the D435if's limitations may lead to the cucumber being invisible. On the other hand, moving too slowly can result in wasted time. A speed between 50 and 70 mm/s is recommended based on this test.

Cutting blades are placed in 3D printed brackets, which are embedded in TCP of the UR3e robot. There should be an appropriate level of tightness between the TCP and the blades to prevent any slipping.

Complementary research is being conducted to determine the mechanical characteristics of the plant. Pressure sensors are being employed to measure the maximum force cucumbers can

withstand, determining their tolerance before permanent damage. This study employs two types of sensors: a finger type, utilized for cutting and gripping forces when applied parallel to the cucumber cross-section, and a plate type, employed for examining the length and curvature effects of cucumbers with applied force normal to the cross-section. The characteristics of vines are also being studied. This data is to be utilized for future automation studies, and this information is included to illustrate that a 360° approach is being taken to develop automation solutions.

2.2 Cucumber Detection Using Select Standard Methods

For accurate object detection, pretrained models have become a common practice, as indicated in the literature review. Pretrained models are neural networks that have undergone training on extensive datasets for general tasks, such as image classification. The pretrained model excels in extracting high-level features from images, enabling the detection of complex patterns and objects. They significantly reduce training time compared to training a model from scratch. The bounding box method is the most prevalent approach to fruit detection. It entails creating a rectangle around the detected cucumbers. Different models in YOLOv8 were applied to detect cucumber bounding box – YOLOL (YOLO Large model), and YOLON (YOLO Nano model). These configurations differ in model size and complexity, allowing users to choose a version that aligns with specific trade-offs between computational efficiency and detection accuracy. The computer vision annotation tool (CVAT) was used to label images for image processing (Fig. 8).



Figure 8: An example of labeled cucumbers using CVAT.

The object detection task began with camera calibration and data preparation. To fulfill the object detection task, two distinct custom datasets were used. The Artificial Cucumber Greenhouse dataset includes synthetic representations of cucumber greenhouses. The second dataset, known as the Real Cucumber Greenhouse dataset, includes images captured within real cucumber greenhouses. Labeling involved outlining rectangles around the cucumbers in the datasets before commencing the model training process. For the cucumber pose estimation, three key points for each cucumber: top, bottom, and middle points are labelled.

YOLOL and YOLON are utilized for bounding box detection. The difference between them lies in their performance characteristics. YOLOL is known for its accuracy but demands more processing time and a more robust computer. On the other hand, YOLON is a lighter and faster model, but typically sacrifices some accuracy compared to YOLOL.

Precision, recall, and the F1 scores are fundamental metrics used in evaluating object detection models, providing insights into their performance regarding true positive (TP), false positive (FP), and false negative (FN) predictions. Precision, denoted as the ratio of TP to the sum of TP and FP, measures the model's accuracy in correctly identifying positive instances. Recall, calculated as the ratio of TP to the sum of TP and FN, assesses the model ability to capture all relevant positive

instances. The F1 score, which varies between 0 and 1, represents the harmonic mean of precision and recall, offers a balanced assessment of a model's overall performance by considering both false positives and false negatives.

$$Precision = \frac{TP}{(TP + FP)} \quad (1)$$

$$Recall = \frac{(TP)}{(TP + FN)} \quad (2)$$

$$F1 = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \quad (3)$$

MATLAB is employed for geometry analyses, utilizing the Curve Fitting Toolbox.

The percentage of cucumbers visible in each image is shown in Fig. 9. Based on this example image, most of the cucumbers are 100% visible. However, the camera moves at different angles and positions and stops whenever it detects a cucumber. A cucumber might be hidden from one angle because it is recessed behind leaves, vines, and blossoms, and there can be effects such as shadows and occlusions that obfuscate the visibility. This is a greenhouse environment; therefore, wind and weather considerations do not need to be addressed. Cucumbers are revealed when the camera shifts to a new viewpoint.

2.3 Proposed Cucumber Detection and Decision-making

Different software tools were employed to detect and characterize the cucumbers. For calibrating the camera, software from Intel is specifically designed for the calibration of D435 series cameras was employed. The Intel RealSense Dynamic Calibrator software involves positioning the printed target in various orientations, directions, and distances for XYZ mode calibration. There is a similar approach for RGB calibration.

A developed Python script processes the data captured by the 3D camera. For the hardware setup, the camera, mounted on the robot's head, can adjust its angle and position to capture images from multiple viewpoints. This enables the detection of cucumbers that are obscured from certain angles due to leaves, vines, and blossoms.



Figure 9: Percentage of visible cucumber in an example image.

A color analysis on the target cucumber and plant components was conducted to establish filter boundaries. After filtering, the target RGB values for cucumbers are kept. Some pixels of cucumber are mistakenly filtered out (false negatives), while some pixels of leaves and vines are inaccurately

identified as cucumber pixels (false positives); therefore, noise cleanup operations needed to be performed.

3 RESULTS AND DISCUSSION

3.1 Cucumber Bounding Box and Key Points Detection Using YOLOv8

The examination of the cucumber bounding box and key point detections is conducted, and select results are showcased. Fig. 10 displays two images in which cucumbers are detected in YOLO model, accompanied by their respective detection confidence for each cucumber. There is no false positive being detected. However, few false negatives were observed, primarily because very small portions of cucumbers are visible in some cases. The percentage of correctness for each detected cucumber bounding box is shown in the figure.

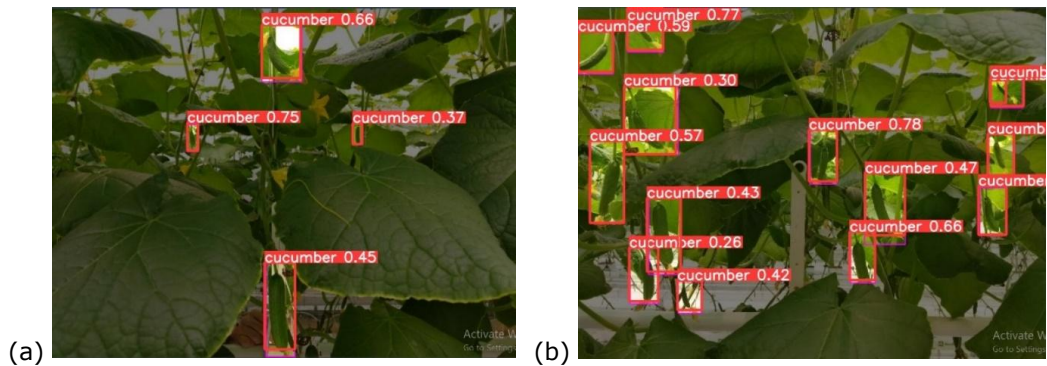


Figure 10: YOLOL- two images with their detected cucumbers and their confidence levels.

Fig. 11 illustrates two images with detected cucumbers with the YOLON approach, along with their corresponding detection confidence levels for each cucumber. Similar to the YOLOL model results, there are no false positives. Nonetheless, few instances of false negatives are noted.

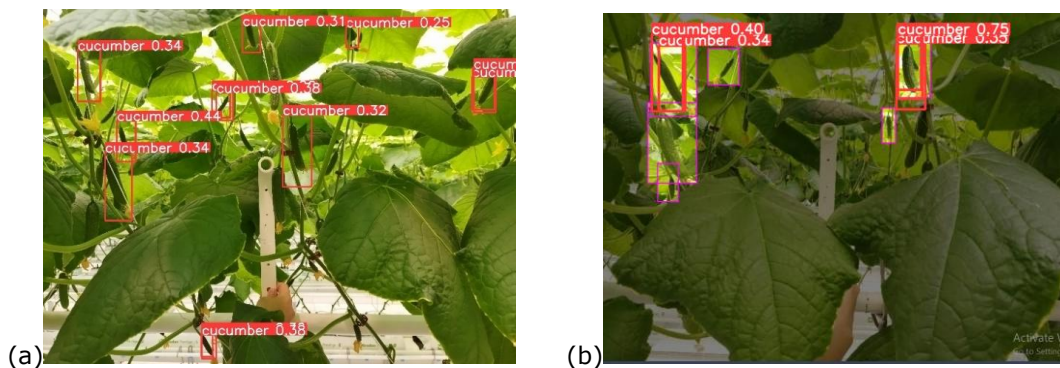


Figure 11: YOLON- two images with their detected cucumbers and their confidence levels.

Two crossing cucumbers in artificial greenhouse are successfully detected using pose estimation model and the results are shown in Fig. 12 (c).

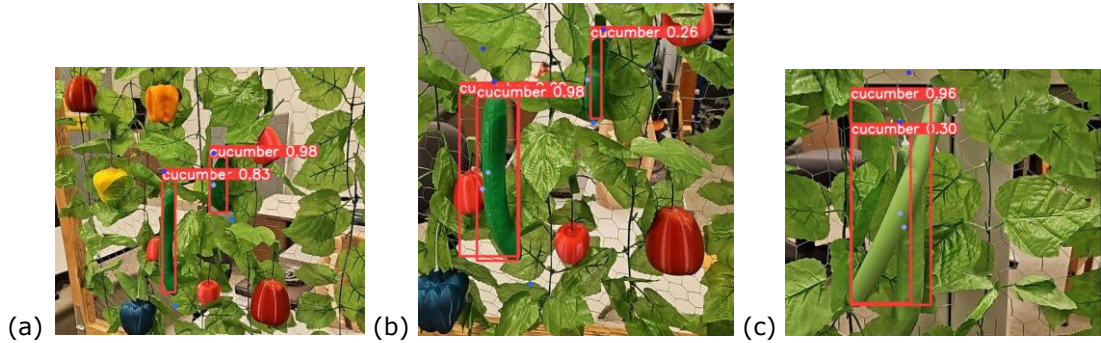


Figure 12: Detected cucumbers and confidence levels for the key points method (blue dots) using the experimental setup: (a-b) unhidden cucumbers, (c) successful detection for two crossing cucumbers.

The summary of the metrics for cucumber bounding box detection and key points is presented in Tab. 2 and 3. YOLOL performed better than YOLON for the bounding box detection, and YOLOL performed better than YOLOX for the key points detection. Scans from multiple angles will improve the bounding box and key points detection, and subsequently improve the performance metrics. However, this dataset is to be input into the determination of a robotic trajectory for harvesting. Depth information is missing. This can be determined with multiple scans and stationary field targets that can be used as a reference.

Model	F1	Confidence*
YOLOL	0.55	0.204
YOLON	0.38	0.121

Table 2: Model and F1 results for the cucumber bounding box detection.

Model	F1	Confidence*
YOLOL	0.82	0.502
YOLOX	0.42	0.053

Table 3: Model and F1 results for the cucumber key points detection.

*The confidence level typically refers to the probability that a predicted bounding box contains an object and how accurate the prediction is.

The image sizes were reduced to approximately one-fourth of their original size captured in the greenhouse. Typically, the original images were around 4 MB, while the YOLO input images were reduced to about 1 MB. This reduction may result in some loss of information, like the detection rate. It is likely that one reason for missed cucumber detections is the image downscaling, as YOLO is not designed to process large images. Further investigation is needed to confirm this and to determine the suitability of YOLOL versus YOLOX.

3.2 Employing Raw RGB-XYZ Data

For the first step, a color analysis of the target cucumbers and plant components is done. Color analysis was conducted for the vines, leaves, stems, and other plant components to establish filters

(Fig. 13). RGB value intervals for type 1 and type 2 of cucumbers (cu), stem ends (st), vines (vi) and leaves (le) are shown in Tab. 4 and 5.

It can be seen that all cucumbers, stem ends, vines, and leaves exhibit almost identical blue values. Therefore, the blue channel does not prove to be effective for detection purposes. Therefore, rules were determined and applied, such as:

Rule 1: Keep R < 85 and remove all R > 85

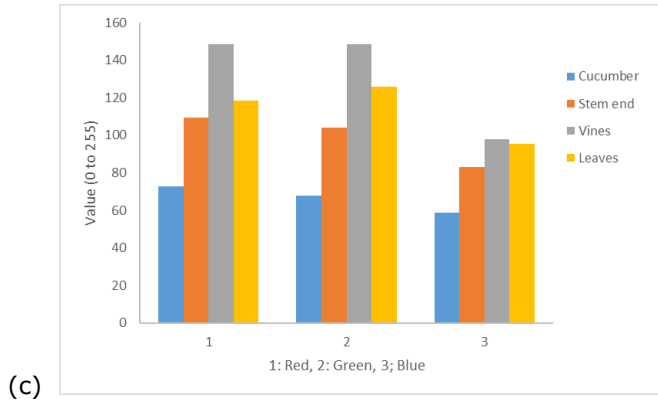
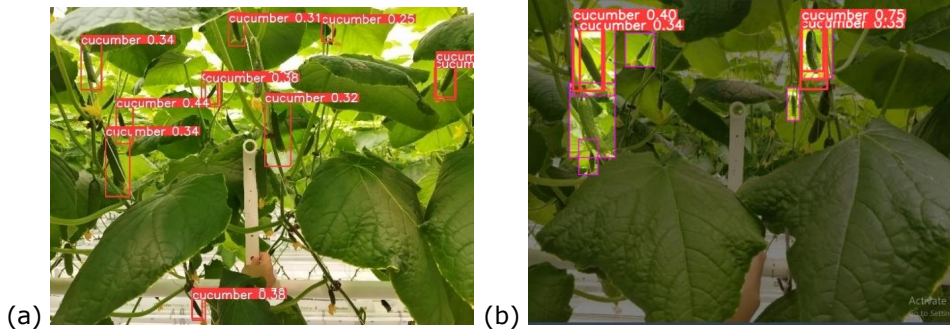


Figure 13: Two different types of cucumbers are grown in JEM farms due to crop rotation. The cucumber on the left (a) is referred to as "type 1," while the one on the right (b) is "type 2." (Courtesy of JEM farms). (c) a graph with observed average values for each channel.

	Red	Green	Blue
Cucumber	61.6<Rcu<81.2	55.7<Gcu<74.7	44.1<Bcu<69.9
Stem end	62.3<RSt<109.2	61.4<Gst<104	39.7<Bst<83
Vine	105<Rvi<148.7	115.4<Gvi<148.6	39.2<Bvi<97.8
Leaf	91.4<Rle<122.8	112.4<Gle<151	43.7<Ble<95.5

Table 4: RGB Intervals for Type 1 cucumbers.

	Red	Green	Blue
Cucumber	42.8<Rcu<88.3	52.2<Gcu<108.9	7.6<Bcu<36.3
Stem end	73<Rst<139.6	98.3<Gst<158.8	8.9<Bst<78.2

Vine	$90 < R_{vi} < 151.5$	$115.4 < G_{vi} < 167.2$	$17.9 < B_{vi} < 50.2$
Leaf	$70.4 < R_{le} < 123.1$	$107 < G_{le} < 148.4$	$14.9 < B_{le} < 53.9$

Table 5: RGB Intervals for Type 2 cucumbers.

This eliminated vine data with an approximate accuracy of 90%. Specific rules were developed for each cucumber type and applied. After filtering, the target RGB values for cucumbers are kept. Some pixels of cucumber are mistakenly filtered out (false negatives), while some pixels of leaves and vines are inaccurately identified as cucumber pixels (false positives), as shown in Fig. 14. However, through proximity analyses to address false positives and false negatives, noise was removed, and gaps were filled.

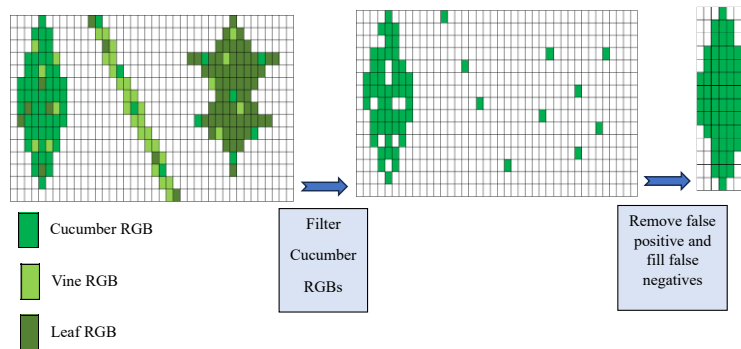


Figure 14: Left: visualization of target RGB values for cucumber, vine, and leaf (each square assumed to be a pixel). Middle: false positives and false negatives pixels. Right: after removing false positives and retaining false negatives, a complete cucumber shape is maintained, while all leaves and vines are effectively filtered out

Once the elements are identified by color and positional data, the XYZ point cloud data is leveraged. With the Z data, the distance between the robot and the cucumber is determined. The pixels are sorted Python. The pixels with minimum distance are chosen as the cutting point if the camera starts scanning from the top because they show the top of the cucumber, which is the point closest to the stem end, as shown in Fig. 15. In this study, it is considered as the initial cutting point. More accurate results and tests are needed to determine the final cutting point (i.e., close to the cucumber body or is some stem acceptable).

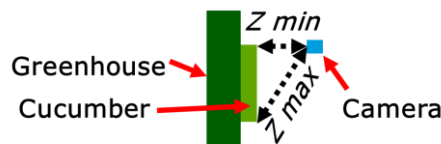


Figure 15: Minimum Z pixel is the cutting point for the cucumber.

3.3 Cucumber Harvesting Test

In an experimental study, the feasibility of the method is tested using a R-R-URK-UR3e robotic arm as shown in Fig. 16. Once the location of the stem end is extracted, it is programmed to move toward the stem end and cut the stem end. The cucumber was then placed into a container. These steps are shown in Fig. 17 (a-e). Several tests were performed, showing the feasibility of the method.

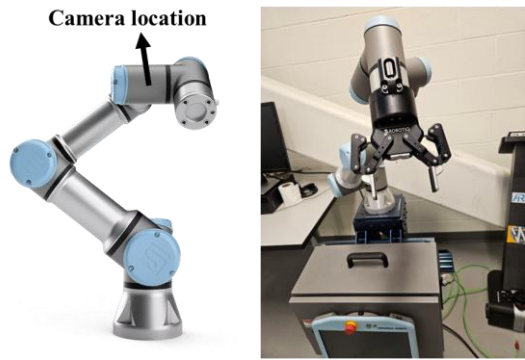


Figure 16: R-R-URK-UR3e robotic arm used for the cucumber harvesting.

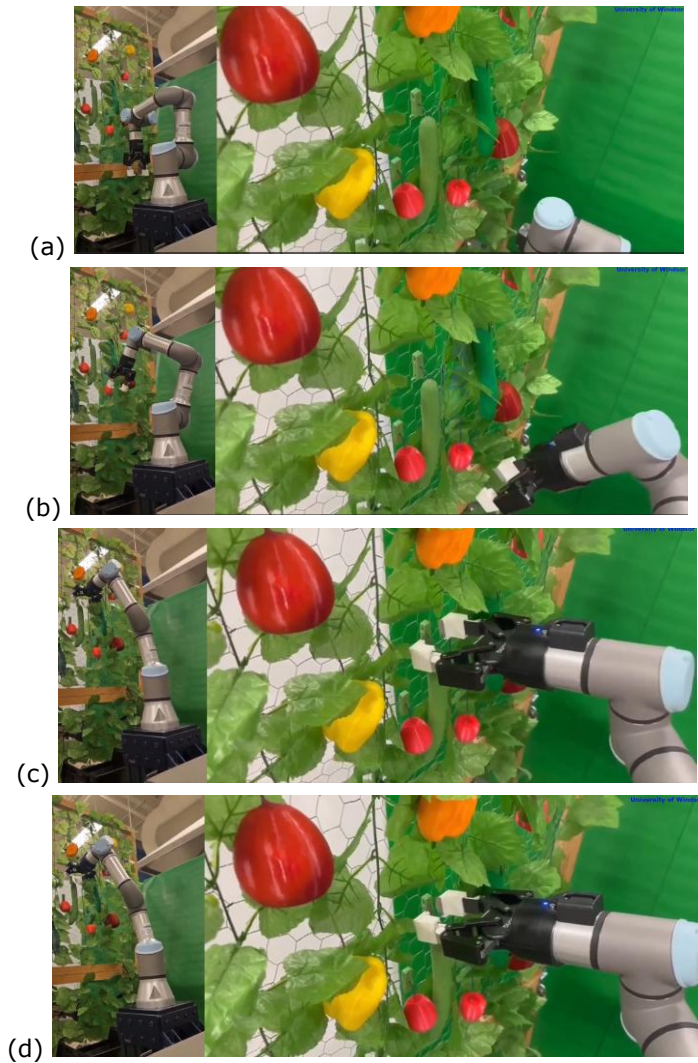




Figure 17: (a – e) Robot trajectory to harvest detected cucumbers based on received position data in a 3D printed artificial greenhouse

3.4 Discussion

There are several image detection and classification algorithms presently being employed in several diverse domains. When manipulating image data for identifying cucumbers, YOLO was the superior model as confirmed by the F1 metric and the fact that there was no false positive when detecting cucumbers. However, there are limitations to this method, in particular finding the key points in Cartesian space for input into a robotic based harvesting system. The proposed cucumber detection method, which uses XYZ-RGB data gained directly from 3D camera, has the potential of augmenting the traditional cucumber detection methods which required time-consuming processes such as preparing an extensive dataset and using neural networks. It is laborious to annotate a diverse dataset. For a robust solution, a comprehensive set of variations of the object size, position, and background must be considered. Once the model is set up (number of classes and other parameters) pre-processing or training must be performed and validated. Data augmentation techniques may be required to help the model generalize better. However, solutions for one greenhouse may not be transferable to another environment. But with a hybrid data collection and analysis strategy where a 3D camera is employed as the data collection tool, RGB analysis can be performed to identify produce, stems, leaves, etc., and XYZ data analyses can be used to create labels at key positions, automating the target creation necessary for contemporary image analyses. The same data sets can be applied to produce quality or 'grading' activities. There are acceptable and unacceptable levels of curvature, size, and so forth. Essentially reverse engineering strategies can be applied along with color analyses to classify the harvest in real time. The process flow for this proposed data fusion approach is shown in Fig. 18.

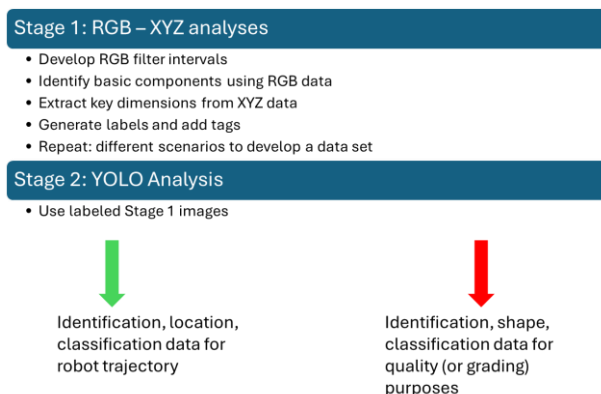


Figure 18: A two-stage image analysis process to automate labeling key features.

Physical experimentation validated that the prototype harvesting framework functioned well for the artificial greenhouse testing, and the cucumber being targeted was harvested successfully. This method could be expanded to different cucumber types, but analyses need to be performed to establish effective filtering rules. Once this has been done, there are several potential implementation strategies that could be applied, which could be as basic as lookup up tables or introducing reinforcement learning strategies to refine and improve rules.

Tests in greenhouse environments need to be conducted during the growth and harvesting cycles in different lighting conditions. Automation can be applied in non-standard working hours and the lighting / shadow conditions vary at night and shadows vary over the year. Another challenge is to develop trajectories that do not touch the leaves (or limit contact). There is significant research yet to be conducted, but fusing point cloud data analyses with contemporary image analyses provides new opportunities to meet these challenges.

With the *XYZ* data, inspection can occur during the harvesting process. Tab. 6 summarizes the grade and associated geometry data for American greenhouse cucumbers. Offline cucumber shape and size analyses were performed from the images captured by the D435if camera. The shape and sizes are effectively determined. Select peripheral data and a central skeleton curve are shown for two samples in Fig. 19. There is less than 5% discrepancy between the measured data and the processed data. This test indicates that sorting and classification activities can be effectively performed by employing the same camera and data set. Future investigations need to consider shading and variable light conditions.

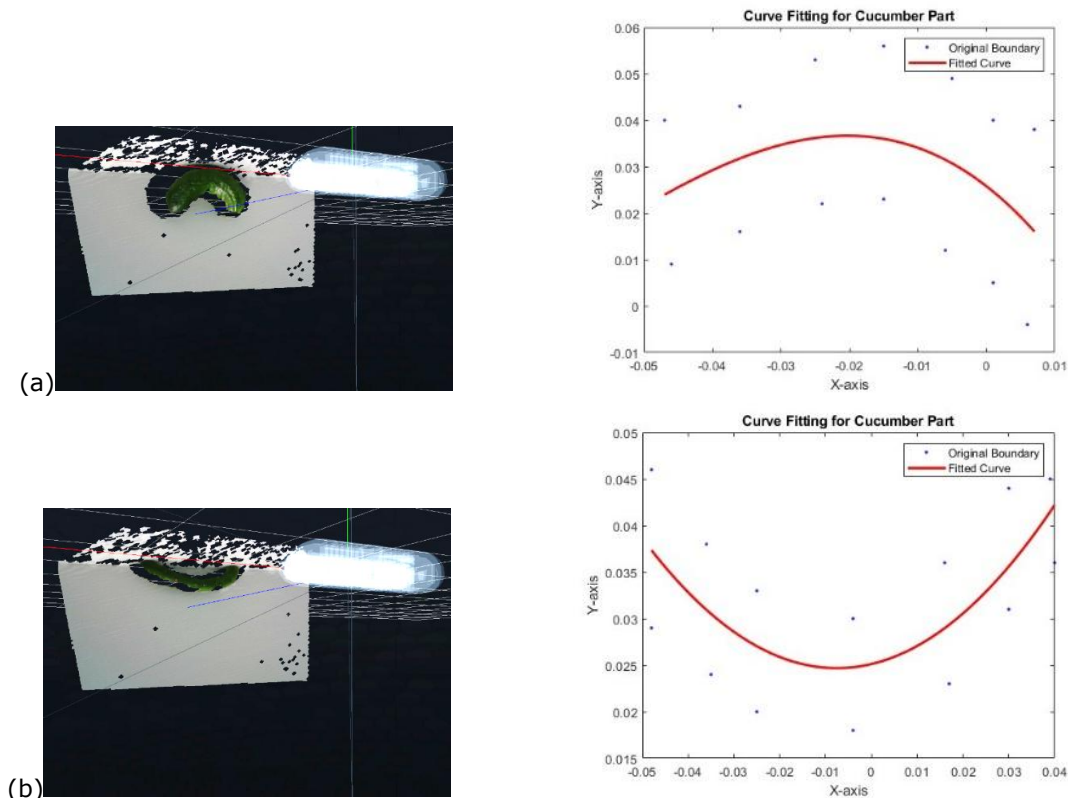


Figure 19: (a) Cucumber 1 image and extracted boundary, (b) Cucumber 2 image and extracted boundary (units are in m).

GRADE	DIAMETER	LENGTH
U.S. Fancy	2-3/8 in. max.	6 in. min.
U.S. Extra No. 1	2-3/8 in. max.	6 in. min.
U.S. No. 1	2-3/8 in. max. UOS*	6 in. min. UOS*
U.S. No. 1 Small	1-1/2 in. min. 2 in. max.	No requirements
U.S. No. 1 Large	2-1/4 in. min.	6 in. min. UOS*
U.S. No. 2	2-3/8 in. max. UOS*	5 in. min. UOS*

Table 6: United States Standards for Grades of Greenhouse Cucumbers.

4 SUMMARY AND CONCLUSIONS

Cucumber harvesting automation strategies are being explored as there are issues with labor availability and the potential of injuries associated with the repetitive tasks being performed in non-ideal postures. A key element of an automation solution is the detection of the cucumbers in 3D space. When utilizing flat images, and tools such as ImageJ, the boundaries are extracted for plant components, but alternative analyses need to be performed to identify the components and isolate cucumbers. YOLO can detect bounding boxes with an appropriately annotated data set, and with multiple scans combined with designed targets, positional data can be extracted. Conversely, employing a 3D camera and analyzing RGB and XYZ point cloud data provides new opportunities. The stem position and shape data can be extracted from the point cloud, enabling a robotic arm to 'snip' a cucumber in the artificial test setting and realistic cucumber shape data can be extracted using reverse engineering approaches. It is proposed to continue to utilize 3D cameras along with reinforcement learning algorithms to improve plant component detection, not only for developing robot trajectories for harvesting, but for collision avoidance (avoiding leaves, and other features/infrastructure).

5 FUTURE STUDIES

The recommended future research work is:

- Process the raw camera data to accurately reconstruct the cucumber's shape to determine whether it should be harvested and use the color variation analyses to indicate probable damaged regions.
- Develop a completely automated solution with the hardware and software fully integrated.
- Perform data collection to capture the characteristics of cucumbers and leaves in real greenhouse environments during the growth cycle. Understanding factors such as common cucumber growth locations, angles for the stems, etc. and where they are typically hidden will help optimize camera movement, improving detection from the initial stage.
- Perform tests under different lighting conditions. An automated system can be employed 24-7, and the lighting conditions will vary depending on the amount of real and artificial lighting.
- Alternative platforms like RoboFlow could be explored to compare their detection results with those obtained from YOLO and XYZ-RGB data management methods. A preliminary study successfully detected cucumbers using RoboFlow, but additional images are needed to enhance the dataset. It is hypothesized that a hybrid solution will provide the most robust solution.

Shahryar Ghorbanirezaei, <https://orcid.org/0009-0006-3091-784X>
 (Ruth) Jill Urbanic, <https://orcid.org/0000-0002-2906-7618>

REFERENCES

- [1] Chang, Y.-K.; Zaman, Q.; Farooque, A.; Schumann, A.-W.; Percival, D.-C: An automated yield monitoring system II for commercial wild blueberry double-head harvester, *Computers and Electronics in Agriculture*, 81, 2012, 97-103, <https://doi.org/ledproxy2.uwindsor.ca/10.1016/j.compag.2011.11.012>
- [2] Crop Profile for Greenhouse Cucumbers in Canada, Pesticide Risk Reduction Program and Pest Management Centre and Agriculture and Agri-Food Canada, Canada, 2006
- [3] Davidson, J. R.; Silwal, A.; Hohimer, C.-J.; Karkee, M.; Mo, C.; Zhang, Q.: Proof-of-concept of a robotic apple harvester, 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016, 634-639, <https://doi.org/10.1109/IROS.2016.7759119>
- [4] Eizentals, P.; Oka, K.: 3D pose estimation of green pepper fruit for automated harvesting, *Computers and Electronics in Agriculture*, 128, 2016, 127-140, <http://dx.doi.org/10.1016/j.compag.2016.08.024>
- [5] Ghorbanirezaei, S.: Greenhouse Cucumber Detection and Characterization for Harvesting Framework Implementation, University of Windsor, Windsor, Ontario, Canada, Thesis, 2024
- [6] Gené-Mola, J.; Vilaplana V.; Rosell-Polo, J.-R.; Morros, Josep-Ramon; Ruiz-Hidalgo, J.; Gregorio, E.: Multi-modal deep learning for Fuji apple detection using RGB-D cameras and their radiometric capabilities, *Computers and Electronics in Agriculture*, 162, 2019, 689-698. <https://doi.org/10.1016/j.compag.2019.05.016>
- [7] Iskandar, D.: Classification of Melinjo Fruit Levels Using Skin Colour Detection With RGB and HSV, *Journal of Applied Engineering and Technological Science*, 4(1), 2022, 123-130
- [8] Intel® RealSense, Intel® RealSense™ Depth Camera D435if, USD, 2025
- [9] Jing, W.; Leqi, W.; Yanling, H.; Yun, Z.; Ruyan, Z.: On Combining DeepSnake and Global Saliency for Detection of Orchard Apples. *Appl. Sci.*, 11, 2021, 6269. <https://doi.org/10.3390/app11146269>
- [10] Kheiralipour, K.; Abbas Pormah A.: Introducing new shape features for classification of cucumber fruit based on image processing technique and artificial neural networks, *Journal of food process engineering*, 40(6), 2017, <https://doi.org/10.1111/jfpe.12558>
- [11] Khurram Monir Rabby M.-D; Chowdhury B.; Kim J.-H.: A Modified Canny Edge Detection Algorithm for Fruit Detection & Classification, 10th International Conference on Electrical and Computer Engineering, 20(22), 2018
- [12] Liu, X.; Zhao, D.; Jia W.; Ji, W.; Sun, Y.: A Detection Method for Apple Fruits Based on Colour and Shape Features, *Computer Science*, 7, 2019, 67923-67933, <https://doi.org/10.1109/ACCESS.2019.2918313>
- [13] McGuinness, B. -J.; Duke M.-D.; Au, K.-C.; Lim, H.-S.: Field factory for the automated harvesting of forestry tree stock, *Biosystems Engineering*, 227, 2023, 52-67. <https://doi.org/10.1016/j.biosystemseng.2023.01.003>
- [14] Molada-Tebar, A.; Verhoeven, G.-J.; Hernández-López, D.; González-Aguilera, D.: Practical RGB-to-XYZ Color Transformation Matrix Estimation under Different Lighting Conditions for Graffiti Documentation, *Sensors*, 24(6), 2024, 1743, <https://doi.org/10.3390/s24061743>
- [15] Shaprov, M.-N; Sedov, A.-V.; Sedova, O.-P.; Ulyanov, M.-V.; Gurba, A.-V.: IOP Conf. Series: Earth and Environmental Science 965, 2022, 012057, <https://doi:10.1088/1755-1315/965/1/012057>
- [16] Statistical Overview of the Canadian Greenhouse Vegetable and Mushroom, Horticulture Section and Crops and Horticulture Division and Agriculture and Agri-Food Canada, Canada, 2024
- [17] Tu, S.; Pang J.; Haofeng Liu H.; Zhuang, N.; Chen, Y.; Zheng, C.; Wan, H.; Xue Y.: Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images, *Precision Agriculture*, 21, 2020, 1072-1091, <https://doi.org/10.1007/s11119-020-09709-3>.

- [18] Universal Robots e-Series User Manual, Universal Robots, USA, 2022
- [19] Wang, W.; Liu, W.; Hu, J.; Fang, Y.; Shao, Q.; Qi, J.: GraspFusionNet: a two-stage multi-parameter grasp detection network based on RGB-XYZ fusion in dense clutter, *Machine Vision and Applications*, 31(7-8), 2020
- [20] Zheng, B.; Sun, G.; Meng, Z.; Nan, R.: Vegetable Size Measurement Based on Stereo Camera and Keypoints Detection, *Sensors*, 22, 2022, 1617. <https://doi.org/10.3390/s22041617>
- [21] Yoshida, T.; Onishi, Y.; Kawahara T.; Fukao T.: Automated harvesting by a dual-arm fruit harvesting robot, *ROBOMECH Journal*, 9(19), 2022. <https://doi.org/10.1186/s40648-022-00233-9>

6 NOMENCLATURE

Annotation	Adding labels or marks to images
Augmentation	Adding various transformations, such as rotation and scaling to an image
Bounding box	A rectangular drawn around a target object
Classification	Assign categories to different images
Collaborative robot	Robots that work human
Color threshold	Setting and filtering special pixels to isolate a target
CVAT	Computer Vision Annotation Tool for annotating images
Key points	Some points by which a pose will be detected
R-CNN	Region-based Convolutional Neural Network
RGB parameters	Red, Green, Blue values of a pixel
RoboFlow	A platform for object detection
UR3e robot	Universal Robots 3e
YOLO	You Only Look Once; a pretrained object detection algorithm
YOLOL	YOLO Large model
YOLON	YOLO Nano model