Computers and Electronics in Agriculture 192 (2022) 106562

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Vision systems for harvesting robots: Produce detection and localization

Luis-Enrique Montoya-Cavero [a], Rocío Díaz de León Torres [b], Alfonso Gómez-Espinosa [a], Jesús Arturo Escobedo Cabello [a,*]

[a] Tecnologico de Monterrey, Escuela de Ingenieria y Ciencias, Av. Epigmenio Gonzalez 500, Fracc. San Pablo, Queretaro 76130, Mexico
[b] Tecnologico de Monterrey, Escuela de Ingenieria y Ciencias, Av. Eugenio Garza Sada 300, Lomas del Tecnologico, 78211 San Luis Potosi, Mexico

ARTICLE INFO

Keywords:
Harvesting robots
Machine vision
Produce detection
Produce localization
Machine learning
Deep learning

ABSTRACT

In the last few years, the use of artificial intelligence and technological advances such as deep learning, new 3D-capable sensors, and edge computing embedded systems have increased produce detection and localization performance for harvesting robots. Unfortunately, this performance increase often requires large datasets that must be manually labeled, large periods for training, increased processing time and power for inference, and a high cost of powerful hardware to run the detection models. This work focuses on providing up-to-date information regarding the state of harvesting robots' vision subsystems, focusing on produce detection and localization research with special attention to the new technology that is being used. A description and analysis of the challenges of introducing this technology to produce detection and localization methodologies are also present in this review. Finally, future trends for harvesting robots' vision subsystems are described and discussed.

1. Introduction

The agriculture industry is currently facing a constant decrease in the workforce (Onishi et al., 2019). Skilled workers that accept repetitive tasks in harsh greenhouse climate conditions are scarce. This reduction of workforce results in a loss of production capacity and an increase in cost. On the other hand, there are important concerns regarding the use of resources for agriculture. Specifically, the long-term impact of soil and water supplies. Because of this, there has been a major interest in Precision Agriculture (PA), as it has been shown it can maximize resources and increase crop yields while minimizing waste, losses, and input costs (Yost et al., 2017). A possible solution to this labor shortage and optimization of resources is the automation and mechanization of tasks performed in production environments (Astill et al., 2020). Such as using harvesting robots to reduce the labor involved in managing crops while still maintaining quality in a shorter period (T. Zhang et al., 2020). Because of this, research in the development of harvesting robots for specialty crops automation has gained traction in the last years (Arad et al., 2020). Specialty crops are characterized by being very labor-intensive throughout their growing cycle: production, harvesting, and processing (Astill et al., 2020).

A complete harvesting robot system must be able to traverse its environment, detect produce, distinguish between ripe and unripe produce, localize, harvest, and store it; all this without damaging both the crop and the produce (Fig. 1) (Astill et al., 2020). To perform these actions, most harvesting robots across the literature have three main subsystems: a vision subsystem that enables the robot to detect/localize the crops and its environment, a motion delivery subsystem that allows the robot to reach the ripe produce, and an end-effector that allows the robot to harvest the produce from its plant without damage (T. Zhang et al., 2020).

Moving the end-effector towards the location of the produce and performing the harvesting action are challenging for harvesting robots as the produce and the crop canopy often vary in shape, color, and size (Barth et al., 2016; Ceres et al., 1998). The produce is also often surrounded by obstacles such as other plant parts, other produce, or infrastructure (support wires, etc.). Therefore, high-performance vision systems capable of adjusting to variable scene conditions for detecting and localizing ripe produce are crucial for harvesting robots (Arad et al., 2019). As they will acquire, process, and output the information that

Abbreviations: CCD, Charge Coupled Device; CMOS, Complementary Metal Oxide Semiconductor; CNN, Convolutional Neural Network; DL, Deep Learning; DoF, Degrees of Freedom; FLIR, Forward-Looking Infrared; KNN, K-Nearest Neighbors; ML, Machine Learning; MS-FRCNN, Multiple Scale Faster Region-based Convolutional Neural Networks; MSX, Multi-spectral Dynamic Imaging; R-CNN, Region-Based Convolutional Neural Network; RGB-D, Red Green Blue and Depth; SSD, Single Shot Multibox Detector; SVM, Support Vector Machine; ToF, Time of Flight; YOLO, You Only Look Once.

* Corresponding author.
E-mail address: arturo.escobedo@tec.mx (J.A. Escobedo Cabello).

https://doi.org/10.1016/j.compag.2021.106562
Received 17 February 2021; Received in revised form 5 November 2021; Accepted 7 November 2021
Available online 22 November 2021

Fig. 1. Tomato, red pepper, and cucumbers are grown in a greenhouse production environment. Centro Agropecuario Experimental del Tecnológico de Monterrey (CAETEC). Grown produce has multiple shapes, colors, and size variations even from the same crop. Ripe tomatoes and red pepper can be identified by performing color analyses. However, the difficulty increases when the produce has the same color as other parts of the crop. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

will allow for the successful planning and movement of the motion delivery system to harvest (Bac et al., 2016).

Because of this, current harvesting robot literature has mostly focused on improving the detection and localization performance by using different sensors, detection methodologies, faster processors, new communication technologies, and even positioning the vision sensor closer to the end-effector. This trend continues today largely influenced by the advances in the field of artificial intelligence. Consequently, the produce detection accuracy has generally improved even at difficult scenes (varying lighting and high occlusions) when compared to simpler techniques such as color threshold (Zhao et al., 2016; Lin et al., 2020b) or texture matching (Kapach et al., 2012). Unfortunately, these techniques often require higher computational power (GPUs) as well as large datasets of the produce (Birrell et al., 2020; Williams et al., 2020; Lin et al., 2019b; Lin et al., 2019a). Fortunately, the recent availability of new hardware such as edge AI embedded systems (Mazzia et al., 2020) and new sensors (Fu et al., 2020) are allowing for more compact and efficient designs without compromising produce detection accuracy.

There are multiple works that describe the state of harvesting robots research. Gongal et al. (2015) reviewed the type of sensors that were being used at the time for produce detection and localization, as well as some of the most popular image classification methods for produce detection. Others have focused specifically on the sensors available for harvesting robots, analyzing their advantages and limitations as well as the detection techniques that are used with each of the sensors (Zhao et al., 2016; Fu et al., 2020). Additional works have focused their efforts on analyzing the applications and impact of artificial intelligence in the field of agriculture (Oliveira et al., 2021; Kamilaris & Prenafeta-Boldú, 2018; Koirala et al., 2019; Rehman et al., 2019). However, a gap regarding the current state of harvesting robots' vision systems and the use of new hardware and artificial intelligence methodologies was identified. Thus, in this work, an analysis of state-of-the-art vision systems for harvesting robots is presented. Focusing on the performance improvements that advances in the last couple of years in artificial intelligence have brought. Analyzing the benefits and limitations of new hardware to identify current and future trends. The following sections provide a brief overview of the overall harvesting robots process, describing the environments and the crops they work with, the harvesting process, and the robot's subsystems. Going into detail about the advantages and limitations of the most common vision system's sensors and their detection and localization methodologies. Section 2 describes the concept of harvesting robots, the environment, and crop characteristics, the complete harvesting cycle, and the most common hardware architectures (subsystems) that are being used across current harvesting robots. Section 3 introduces the most widely used 2D and 3D-capable vision sensors as well as multisensory systems for harvesting robots, analyzing their advantages and limitations. Section 4 introduces the concept of feature extraction. Discussing the most common features that are extracted depending on the sensor's capabilities. It then analyzes the most common detection methodologies to process features extracted from the sensor information to detect ripe produce in a scene. Section 5 discusses the most widely used techniques for computing the 3D spatial location of the produce with respect to the robot or the vision system's sensor. Section 6 introduces the concept of eye-hand coordination, in which the vision system and the robot's motion delivery system work together as a control system for produce harvesting. Section 7 analyzes the current trends that harvesting robot development is following; discussing the research hot spots that show the possibility of an increased harvesting success. And finally, conclusions are drawn in Section 8.

2. Harvesting robots

The main objective of a harvesting robot, as its name implies is to obtain ready-to-harvest produce. To do so, a harvesting robot requires multiple sensors to sense its environment. It then uses the acquired information to perform harvesting actions (Edan and Miles, 1994). This is a very complex process that requires multiple subsystems to work together. On top of that, a harvesting robot is also expected to work in multiple environments in which produce is grown, while also working with many different types of crops that differ in color, size, and shape.

This section introduces and analyzes the environment's characteristics in which harvesting robots are expected to work. It then introduces the produce harvesting cycle going into detail about the steps that a harvesting robot requires. Finally, the most widely used harvesting robot subsystems are introduced: (1) The vision subsystem which allows the robot to sense its environment (for autonomous navigation) and/or the crop (for harvesting), (2) the motion delivery subsystem, that allows the robot's end-effector to reach the target produce, and (3) the end-effector subsystem which allows the robot to touch and manipulate the produce for harvesting.

2.1. Environment and crop

Most specialty crops are grown in four main production environments: orchards, greenhouses, indoors, and open fields. Large plants such as trees are grown in orchards, whereas smaller plants are grown in greenhouses, indoors, and in open fields (Bac et al., 2014).

Production environments influence the robot's harvesting performance by introducing multiple factors such as weather or lighting conditions that the robot needs to cope with. Outdoor production environments typically are more difficult for harvesting robots, as there will not be protection against wind or rain, lighting is uncontrollable, and accessibility of objects might be limited if the robot is not capable of traversing rough terrain (e.g., slopes). Whereas in indoor production environments there is protection against rain and wind, lighting can be controlled by the use of artificial lighting, and the terrain is easier to navigate (e.g. greenhouses typically separate rows of crops by fixed-width flat isles) (Bac et al., 2014).

Moreover, parts of the plant such as the fruits, leaves, or branches vary significantly in shape, position, size, and reflectance even for the same crop (Bac et al., 2014). In Fig. 1, the positions of the produce are also widely distributed across the crop's canopy which can sometimes produce occlusions due to the overlapping of the same produce or from

leaves and branches of the crop. Furthermore, the shape and color of the produce vary significantly. As the height/width ratio is not constant even within the same crop (Bac et al., 2014).

These characteristics make the process of harvesting very difficult because a harvesting robot must be able to cope with the crop and environment variables on top of also being able to detect, localize, and harvest the produce. Not to mention that the harvesting robot must also be able to identify individual produce even when it is overlapping. Because of the large variation in produce shape, color, and sizes, most harvesting robots are purpose-designed to work in a specific production environment with a specific crop. Effectively reducing the variables that directly influence the harvesting robot performance (e.g., a robot working at a greenhouse does not need to worry about rain; wind and lighting can be controlled). In other words, the harvesting robot's subsystems will only be able to accommodate one type of crop. This will be discussed in depth in Section 2.3.

2.2. Harvesting process

The harvesting process for specialty crops requires that workers can accurately distinguish individual ripe produce and gently pick, sort, or package the fruit or vegetable by hand without damage (Astill et al., 2020). Broadly speaking, this process also applies to harvesting robots. Fig. 2 shows the high-level harvesting robot process. First, the robot is required to navigate across the production environment, (2) then it needs to detect the produce, process its ripeness, and compute its 3D location; steps which are performed at produce recognition. After this, (3) the robot needs to move towards the detected produce to harvest it and finally (4) store it. All of this, without damaging the crop and the produce.

To perform all these actions, researchers have experimented with multiple subsystems that allow the robots to perform one or multiple processes described in Fig. 2. These subsystems are described in the following section.

2.3. Harvesting robot subsystems

As discussed, most harvesting robots developed today are purpose-built for a specific crop and production environment. As such, researchers have tackled the harvesting robots process (Fig. 2) tasks by implementing multiple hardware architectures. This behavior can be seen in Table 1, where several state-of-the-art harvesting robots were analyzed by the type of crop and the production environment where they are working, the hardware and software architectures, and the performance that they achieved.

Fortunately, even though multiple hardware architectures exist, there are 3 main subsystems (See Fig. 3) that are present at all harvesting robot configurations: (1) a vision subsystem that enables the robot to detect the crop in its environment, its ripeness and then its 3D location by processing information from a vision sensor (2) a motion delivery subsystem that allows the robot's end-effector to reach the ripe produce, and (3) the end-effector subsystem that allows the robot to harvest the produce from its plant without damage (T. Zhang et al., 2020).

This section introduces the processes that each subsystem must perform and analyses the most common characteristics that were found in recent literature on harvesting robots.

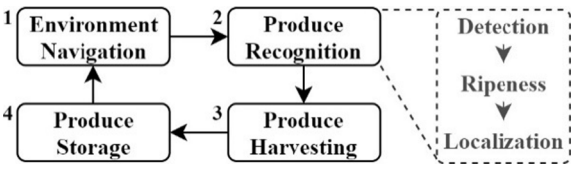

Fig. 2. High-level harvesting robots process. Produce detection, ripeness, and localization are performed at produce recognition.

2.3.1. Vision subsystem

As mentioned, the main objective of a harvesting robot is to harvest ripe produce from a crop. Because of this, vision systems are crucial, as they will provide the information that the other two subsystems require to perform the harvesting action. As a result, the vision system is perhaps the most widely researched subsystem. Detecting ripe produce at production environments is very challenging because of the natural variations of the environment and the crop, and the robot's hardware configuration.

These variables (See Fig. 4) can be classified according to what produces them: (1) Robot variables are factors attributed to the robot position and geometry: vision systems information will be limited to what they can perceive from the robot's position; data noise will also vary a lot if the crop's canopy is moved by the robot's end-effector. (2) Crop factors are related to the physical properties of the produce and the crop's leaves and branches: position, color, texture, size, etc. (3) Environmental factors are related to conditions such as indoor/outdoor locations, wind, rain, lighting, etc., which can impact the performance of a vision system (e.g., lens fogging because of condensation, fruit swaying because of the wind). Fortunately, these factors can be reduced completely or partially if the production environment is indoors: greenhouses are not affected by wind or rain; while lighting conditions and obstacles can be controlled (Bac et al., 2014).

Because of these challenges, researchers have experimented with multiple vision systems, as their sensitivity to environment and crop variables is different. However, after reviewing all projects available in Table 1 the most common components for a harvesting robot vision system are depicted in blue in Fig. 3: the main vision sensor, which is responsible for acquiring data from the crop and/or the environment, and an onboard computer that has ripe produce detection and localization algorithms to process the sensor's information.

Typical industrial machine vision systems often utilize an artificial light source to ensure the same lighting conditions when analyzing objects (Yasmin et al., 2020; Xiang et al., 2020). There are harvesting robots that also incorporate this component, however, most robots do not, as they rely on the lighting conditions of the production environment. This behavior can be seen in the works analyzed in this work (Table 1) as only 2 projects out of 30 used artificial lighting: (Arad et al., 2019) used an onboard flash component, arguing that their system provided robust sweet pepper detection with modest computational resources even on varying lighting conditions.

In Fig. 5, the relationship of the vision system (red outline) with the harvesting robot process (Fig. 2) can be seen. Overall, harvesting robots' vision systems are designed to detect and localize ripe produce. To do so, most vision systems perform two separate processes: (1) produce detection, which processes the vision system sensor's data to first detect produce and then its ripeness (Section 4); and (2) localization, a process which can compute 3D (3 dimensional) produce coordinates with respect to the vision system sensor (Section 5). The information produced in this last step is then used by the motion delivery subsystem to compute the position of the end-effector with respect to the produce.

To ensure data accuracy, vision sensors also required an additional step before being used: calibration. This step is typically done at camera-based sensors and it determines optical parameters used for noise removal and/or distortion correction when gathering data (C. Wang et al., 2017).

As seen in Fig. 5, once the camera is calibrated and depending on the type of sensor, it will produce either 2D or 3D information from the scene. This data is then processed by the produce detection phase, which typically outputs 2D bounding boxes of ripe produce. Then, if the sensor can provide 3D information, the localization phase combines the 2D bounding boxes of the detection phase with 3D data to output the 3D location of a produce. Which is later transformed into real-world coordinates and sent to the motion delivery subsystem to harvest ripe produce using its end-effector.

As will be discussed in Section 4, there is a research trend of applying

Table 1
State-of-the-art Harvesting Robot Vision Subsystem Performance.

Crop (Reference)	Main Sensor	Production Env.	Hardware (Specs)	Produce Detection					Produce Localization				Harvesting		Detection/ Localization Limitations
				Methodology	Precision [%]	Recall [%]	F1 Score [%]	Time [s]	Methodology	Pose	Success [%]	Time [s]	Success [%]	Cycle Time [s]	
Apple (Gené-Mola et al., 2019)	3D LiDAR	Orchard	Intel i7-4500U 2.4 GHz	Reflectance threshold, DBSCAN and K-means clustering	NA	82.4	85.8	11.2	2D to 3D transformation		87.5	11.2	NA	NA	Small dataset, occlusions/NA
Apple (Onishi et al., 2019)	Binocular (RGB-D)	Lab Conditions	?/Titan X (?/?)	SSD	100	92.31	96.0	2	2D to 3D transformation		NA	NA	NA	16	NA/NA
Apple (Feng et al., 2019)	Multi-spectral	Orchard	Intel i7 8750H (2.21 GHz)	SVM	95.74	92.69	94.1	0.7	NA		NA	NA	NA	NA	Different temperature due to sunlight/NA
Apple (Yan et al., 2021)	RGB camera	Orchard	Intel i7 9750H/ NVIDIA RTX 2060 (2.6 GHz/6 Gb)	Modified YOLOv5	83.8	91.48	87.4	0.01	NA		NA	NA	NA	NA	Dataset variety and size/NA
Citrus (Lin et al., 2019a)	Active (RGB-D)	Orchard	Intel i7/ NVIDIA GTX 1060 (?/6 GB)	2D Bayes Segmentation, 3D Density Clustering, and SVM	98.3	86.3	91.9	7.9	3D Cluster mean		x: 7 ± 2.5 mm, y: −4 ± 3 mm, z: 13 ± 4 mm	NA	NA	NA	Adjacent fruits were grouped as one/ Depth precision susceptible to strong lighting
Citrus (Gan et al., 2018)	RGB and Thermal Camera	Orchard	NA (?)	Faster RCNN, Hough Transform	96.4	78.4	86.4	NA	NA		NA	NA	NA	NA	Relative humidity, temperature, and wind greatly affect thermal image performance/NA
Citrus (C.H. Yang et al., 2020)	Active (RGB-D)	Orchard	Intel i7 7800X/2 NVIDIA GTX 1080Ti (?/11 GB)	Mask R-CNN	88.15	79.85	83.7	9.2	2D to 3D transformation		H: ± 2.25 V:± 2.29	NA	NA	NA	NA/NA
Cucumber (Mao et al., 2020)	RGB camera	Greenhouse	Intel i7 8550U/ NVIDIA MX150 (1.99 GHz/ ?)	CNN and SVM	91.4	NA	NA	NA	NA		NA	NA	NA	NA	If produce is obstructed, multiple cucumbers are detected/NA
Cucumber (Fernandez et al., 2018)	RGB camera	Field	Intel Xeon E5-1620 (3.6 GHz)	SVM segmentation and morphological operation	85.	90.1	87.8	0.92	NA		NA	NA	NA	1.11	Occlusions and lighting/NA
Eggplant (Sepulveda et al., 2020)	RGB and ToF cameras	Lab Conditions	Intel i7 4790 (3.6 GHz)	SVM	88.3	88.1	87.8	NA	2D to 3D Transform	x		0.81	91.67	26.2	Lighting conditions/NA
Eggplant, apple, grape (T. Zhang et al., 2020)	Active (RGB-D)	Lab Conditions	NA	Mask R-CNN	80	87	NA	NA	2D to 3D transformation		81	NA	70	NA	Occlusions by leaves, stems, and other crops/Failed at irregularly shaped crops
Eggplant, guava (Lin et al., 2020b)	RGB camera	Multiple	Intel i3 4150/NA	Shape matching, probabilistic	greater than78.3	greater than0.754	NA	NA	NA		NA	NA	NA	NA	Overlapping fruits/ NA

(*continued on next page*)

Table 1 (*continued*)

Crop (Reference)	Main Sensor	Production Env.	Hardware (Specs)	Produce Detection					Produce Localization				Harvesting		Detection/ Localization Limitations
				Methodology	Precision [%]	Recall [%]	F1 Score [%]	Time [s]	Methodology	Pose	Success [%]	Time [s]	Success [%]	Cycle Time [s]	
			(3.5 GHz/ NA)	Hough transform, SVM											
Grapes (Kalampokas et al., 2021)	RGB camera	Dataset	NVIDIA RTX 2070/Jetson TX2 (?/?)	CNN	98.1	NA	86.7	8.3/ 2.8	NA		NA	NA	NA	NA	Small dataset/NA
Grapes (Luo et al., 2018)	RGB camera	Greenhouse	Intel i5 3230 M (2.6 GHz)	Color segmentation	81.6	NA	NA	0.69	NA		NA	NA	NA	NA	Changing illumination, occlusions by leaves/NA
Grapes (Luo et al., 2016)	RGB camera	Field and Greenhouse	Intel i5 3230 M (2.6 GHz)	Color Component Classification Cluster separation: Adaboost	96.5	NA	NA	0.59	NA		NA	NA	NA	NA	Change of color due to varying illumination, overlapped grapes/ NA

Crop (Reference)	Main Sensor	Production Env.	Hardware (Specs)	Produce Detection					Produce Localization				Harvesting		Detection/ Localization Limitations
				Methodology	Precision [%]	Recall [%]	F1 Score [%]	Time [s]	Methodology	Pose	Success [%]	Time [s]	Success [%]	Cycle Time [s]	
Grapes and Apples (Fernández et al., 2014)	RGB, Thermal, and ToF Camera	Orchard and Field	NA (?)	SVM	99.8	NA	NA	NA	2D to 3D Transform		x: ± 4.5 cm, y: ± 6.1 cm, z: 7.1 cm	NA	NA	NA	Shadows and specular reflections/ Reflections on odd shape produce
Guava (Lin et al., 2019b)	Active (RGB-D)	Orchard	Intel i5 3210M/ NVIDIA GTX 1060 (?/6 GB)	FCN and Euclidean Clustering	98.3	94.9	96.5	1.39	2D to 3D transformation	x	23.43∘ ± 14.18∘	1.398	NA	1.39	2 fruits clustered as 1 due to sunlight noise in sensor/Failed object detection
Kiwi (Massah et al., 2021)	RGB camera	Orchard	Intel i7 (2.8 GHz)	Optimized SVM	95	80	86.8	0.55	NA		NA	NA	NA	NA	NA/NA
Kiwi (Williams et al., 2020)	Binocular (RGB-D)	Orchard	?/8 NVIDIA GTX 1080 (?/?)	Faster-RCNN 2.0	95	85	90	2.7	2D to 3D transformation		NA	NA	89.3	3.36	Variable lighting/NA
Lettuce (Birrell et al., 2020)	RGB camera	Field	Intel i7 7700k/ NVIDIA 1080Ti (?/?)	YOLOv3	NA	100	NA	NA	Aruco markers 2D to 3D transformation		91	NA	88	31.7	Variable lighting/ Camera calibration
Litchi (C. Wang et al., 2017)	Binocular (RGB-D)	Orchard	Intel i5 2500 (?)	Bayesian, KNN, SVM, ANN	89.1	92.8	90.9	1.36	2D to 3D Transform		NA	2.536	NA	NA	Varying illumination and fruit occlusions
Orange, pineapple, cucumber(N. Guo et al., 2020)	Active (RGB-D)	Lab Conditions	NA(?)	Shape Matching	NA	NA	NA	NA	Shape Matching	x	x: ± 3.121, y: ± 0.777, z: 2.001	12.75	NA	NA	Reconstruction model quality
Passion Fruit(Tu et al., 2020)	Active (RGB-D)	Orchard	Intel Xeon E3 1245/NVIDIA GTX Titan (3.4 GHz/?)	MS - FRCNN	96.2	93.1	94.6	0.17	NA		NA	NA	NA	NA	Occlusions and lighting/NA
Peach(Wu et al., 2020)	Active (RGB-D)	Orchard	Intel i5 6500 (3.2 GHz)		53.16	80.3	63.9	60	2D to 3D transformation		NA	NA	NA	NA	Irregularly shaped fruits/Strong Lighting

(*continued on next page*)

Table 1 (*continued*)

Crop (Reference)	Main Sensor	Production Env.	Hardware (Specs)	Produce Detection					Produce Localization				Harvesting		Detection/ Localization Limitations
				Methodology	Precision [%]	Recall [%]	F1 Score [%]	Time [s]	Methodology	Pose	Success [%]	Time [s]	Success [%]	Cycle Time [s]	
				2D and 3D feature fusion with SVM, VFH with KNN											
Strawberry (Xiong et al., 2019a)	Active (RGB-D)	Greenhouse	Intel i5 6700 (?)	Color Threshold and morphological operations	NA	NA	NA	0.1	2D to 3D transformation		NA	0.11	5–97.1	6.1	Occlusion by own motion delivery system/Limited 3D information
Strawberry(Zhou et al., 2020)	Binocular (RGB-D)	Greenhouse	Intel i5(2.3 GHz)	Faster RCNN	NA	NA	88.9	NA	NA		NA	NA	NA	NA	Fruit angle variation in dataset/NA
Sweet Pepper(Arad et al., 2020)	Active (RGB-D)	Greenhouse	?/GPU(?/?)	Color and morphological operations + SSD	NA	73	NA	NA	2D to 3D transformation + Visual servoing	x	47	NA	61	24	Occlusions by leaves and stems/NA
Sweet Pepper(Chen et al., 2020)	RGB camera	Synthetic Dataset	Intel i5 8400/ NVIDIA Tesla P40	CNN, contour-finding	81.99	98.75	89.5	1.7	NA		NA	NA	NA	NA	Computational performance/NA
Tomato(L.L. Wang et al., 2017)	Binocular (RGB-D)	Greenhouse	?(?)	Color Threshold	NA	50–70	NA	15	2D to 3D transformation		NA	NA	86	15	Overlapping tomatoes and occlusions/Light intensity, overlapping objects calibration errors.
Tomato(Liu et al., 2019)	RGB camera	Greenhouse	Intel i5 ?/NVIDIA GTX 1070Ti (3.3 GHz/?)	Modified YOLOv3	93.0	94.7	93.9	0.05	NA		NA	NA	NA	NA	Occlusions/NA

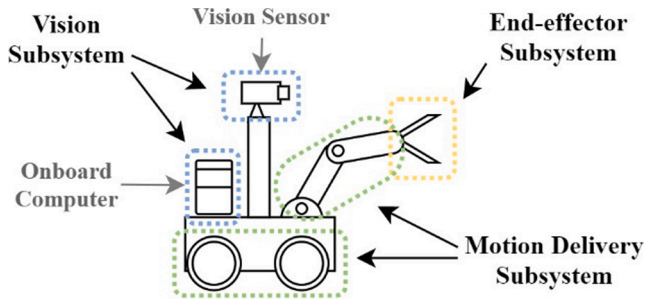

Fig. 3. Main subsystems for harvesting robots. In grey text, harvesting robot vision subsystem components.

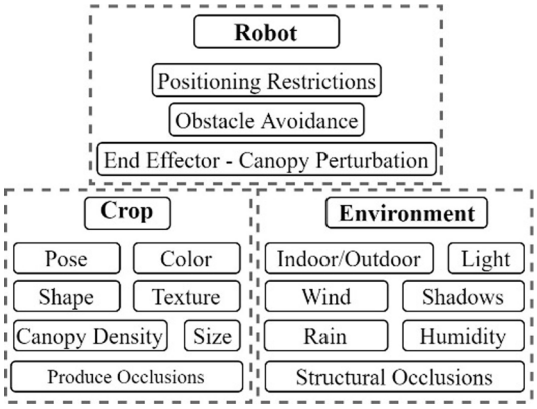

Fig. 4. Variables affecting Vision Systems grouped by source.

deep learning methodologies into the produce detection and localization processes of harvesting robots. As they have consistent results even on challenging scenarios such as varying lighting and multiple occlusions. Unfortunately, these methodologies require powerful hardware such as graphics processing units (GPUs) onboard. As such, harvesting robot vision systems often include a powerful computer with dedicated GPUs. Fortunately, recent technological developments such as Edge AI have allowed for smaller embedded hardware accelerated devices (NVIDIA's Jetson family or Intel's Neural Computing Sticks), capable of achieving real-time inference without compromising on the detection accuracy (Mazzia et al., 2020).

In contrast to the use of Edge AI embedded systems, another possible solution is to connect the vision system sensor to the cloud via a high-speed wireless connection such as 5G. Unfortunately, 5G is yet to be fully implemented and supported in the agriculture industry. However, there is high interest in the benefits that this technology will bring. Especially for big 5G service providers such as Telcel in Mexico owned by América Móvil, who has shown its interest in the applications of Smart Farming (Rodríguez, 2021), such as applying Internet of Things to optimize resources and optimize the yield performance. Other applications of high-speed wireless connection involve the collection, analysis, and use of big data for precision agriculture (Rosales-Soto & Arechavala-Vargas, 2020). This will allow for intelligent decision-making based on real data, will optimize the resources, and produce yield, and will allow for a more competitive market as consumers trends will allow for the anticipation of products to match expected demand. A 5G network permits to perform inference off-site from the production environment. Allowing the vision system to fully exploit the potential of high-performance GPUs on the cloud without compromising on the inference speed (Nasir et al., 2020). GSMA (2020) created a case study for a weed elimination autonomous robot with 5G. The robot captured images at the production environment which were sent to a cloud edge for offsite processing through a 5G connection. Then, the results were used to apply herbicide onsite. This proof of concept demonstrated that powerful computers for model inference will no longer be needed as part of the vision subsystem if a 5G network is available.

While 5G networks have yet to be fully implemented and supported in the agriculture industry, current research shows that a high-throughput wireless connection has multiple benefits. Particularly, because it allows a stable, fast, and reliable connection link between IoT devices and the cloud; for intelligent decision making. Simplifying the information interaction among the equipment being used at the production environment such as robots and machine vision platforms for unmanned farms (Wang et al., 2021).

As mentioned, vision subsystems in harvesting robots are mostly used for autonomous navigation through the production environment

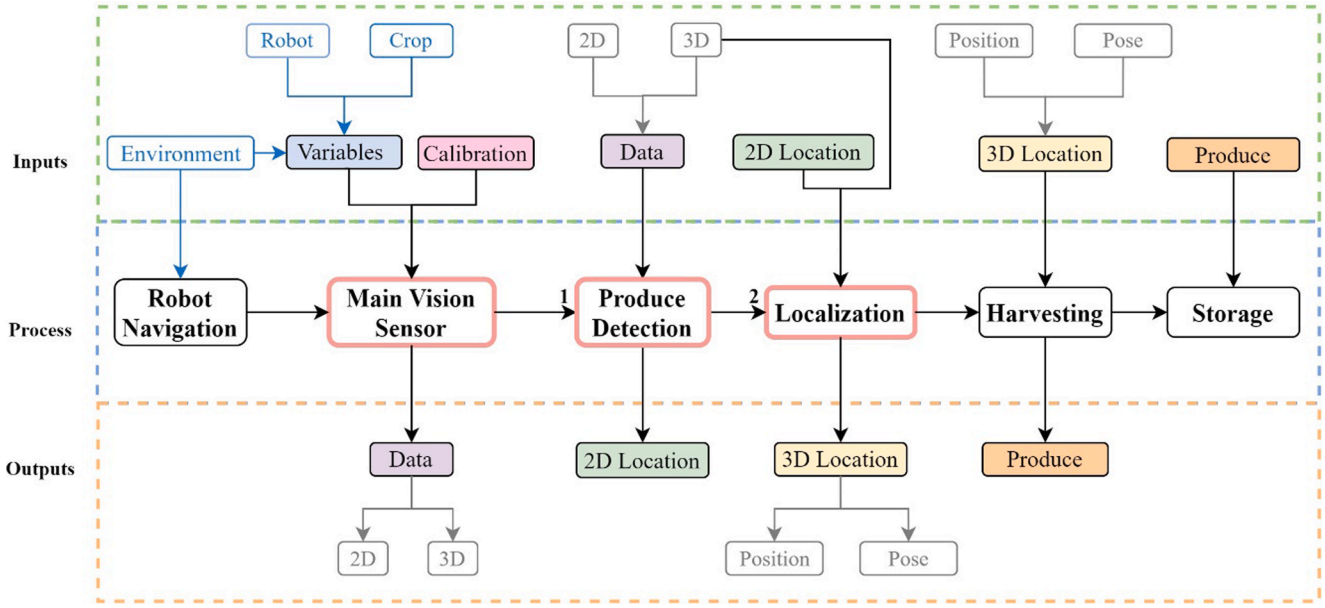

Fig. 5. Vision Systems (red outline) in the harvesting robot's process. Harvesting robots' vision systems consist of a main vision sensor, (1) the produce detection process, and (2) the localization process. Variables affecting vision systems (blue outline). In grey, possible data types. Notice how the output of one process is the input of the next process. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(Le et al., 2020; Zoto et al., 2020) as well as produce detection and localization. However, vision subsystems have many other different use-cases in the agricultural industry. Mavridou et al. (2019) concluded that most agricultural activities that rely on vision subsystems can be grouped in three main approaches: (1) Plant and Food Detection, where the objective is to separate the plant and/or fruit from the rest of the background. Useful for other tasks such as yield estimation, and disease detection. (2) Harvest Support, an activity that involves the gathering of ripe produce. However, also includes activities such as food grading, where produce is sorted based on morphological parameters and maturity level, allowing good quality control. And (3), Plant/Fruit Health Protection and Disease Detection. Which includes activities such as weed, insect, and disease and deficiencies detection (Fountas et al., 2020).

2.3.2. Motion delivery subsystem

Once ripe produce is detected and located, accessing it also presents a challenge. As the robot must avoid damaging other produce, parts of the crop, and support structures from the production environment (Bac et al., 2014). Ensuring that the end-effector reaches the detected ripe produce without damaging other parts of the crop, is the main task of the motion delivery subsystem.

The current most popular motion delivery subsystems in robot harvesting are off-the-shelf industrial anthropomorphic and cartesian robotic arms (See Fig. 6), of which anthropomorphic arms are the most widely used (see Table 2). Typically, researchers combine these industrial robotic arms with a mobile platform that allows traversing the production environment (See Fig. 3). Today, most mobile platforms use a four-wheel-drive motor design with sufficient ground clearance to traverse the production environment where the robot is working at: harvesting robots working at production environments with flat floors do not require much ground clearance such as the ones built by Arad et al. (2020) and Feng et al. (2018), while robots working in open fields do. Because of this, outdoor harvesting robots use all-terrain mobile platforms that have bigger wheels such as the ones used by Birrell et al. (2020), Williams et al. (2020), and Kang et al. (2020) or tracks (Massah et al., 2021) for better mobility.

Anthropomorphic robot arms have the advantage that they replicate the movement of a human arm by joining multiple links with rotational joints. This might be desired, as most of the current production environments were designed to be worked by humans. And as such, a robot capable of understanding and replicating the complete human harvesting process (Fig. 7) might increase harvesting success (Vasconez et al., 2021). This is necessary if harvesting robots are intended to automate the harvesting process in the future. Successfully replicating the human harvesting process has multiple advantages: a harvesting robot does not get tired after long hours of work, and as such, can continue working as long as an adequate power source is provided. Unfortunately, most harvesting robots today rely on batteries as their primary power source, and as such, take considerable time charging. This problem can be solved by having multiple harvesting robots on-site: while some are harvesting, others are charging. Another advantage is that human error can be drastically reduced by introducing harvesting robots: harvesting robots will only harvest ripe produce that successfully matches the characteristics that were programmed, whereas a human harvester might incorrectly harvest unripe produce. As such, harvesting robots' development has focused mostly on precision agriculture environments, where it is necessary to harvest produce individually as not all fruit will ripen at the same time (e.g., tomatoes, sweet pepper, eggplant, etc.)

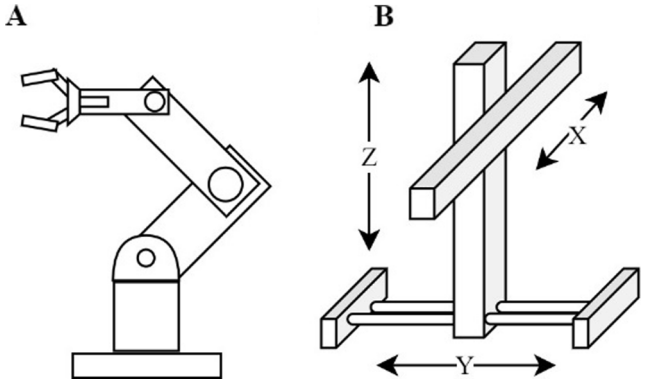

Fig. 6. Most widely used motion delivery subsystem architectures for harvesting robots. (A). Anthropomorphic robotic arm with rotational joints. (B). Cartesian Robotic Arm with linear joints.

Unfortunately, current harvesting robots tend to not harvest produce that is highly occluded. In other words, today's harvesting robots do not perform a search process throughout the canopy while also removing obstacles. Replicating the human harvesting process would require many degrees of freedom (DoF) as well as multiple sensors to ensure the motion delivery subsystem doesn't get stuck. In contrast, using a simpler harvesting process in which the harvesting robot only acquires produce outside of the canopy can be achieved by using cartesian robot arms. This arm architecture can achieve 3D movement by using linear joints. Reducing the required number of DoF at the cost of less maneuverability. This behavior can be seen in the works presented in Table 2, as the robots that have an anthropomorphic arm consistently use >5 DoF, while the robot using a cartesian manipulator only required 3 DoF. Another important observation is that anthropomorphic arms have more consistent harvesting success (lowest was 55%) while the cartesian robot reported a harvesting success range of 5 to 97.1 %, which can be attributed to the maneuverability of the anthropomorphic arms.

While researchers do not justify the selection of the robotic arm architecture, most cartesian-based projects are used for produce that does not have many obstacles and is relatively easy to reach. Xiong et al. (2019a) used this architecture in a table-top strawberry harvesting robot. By design, most ripe strawberries hang in easy to reach positions with almost no obstacle aside from other strawberries. In contrast, anthropomorphic arms are used to harvest difficult-to-reach produce. Arad et al. (2019) and Bac et al. (2016) used this type of robotic arm to harvest sweet pepper. A crop whose produce is typically hard to reach due to occlusions such as leaves and stems. However, a robotic arm that is capable of sensing obstacles when trying to harvest occluded produce, is yet to be developed. As current anthropomorphic arms will consistently get stuck and/or reach a different location if the produce is very occluded because they do not try to maneuver obstacles as a human harvester. Decreasing the overall harvesting success and increasing the harvesting time.

2.3.3. End-effector subsystem

Once the robot reached the target produce, the end-effector is responsible for harvesting it (Hua et al., 2019). This subsystem is mostly purpose-designed to the morphological characteristics of the crop that is being harvested, as produce grows in different shapes and sizes, however, some harvesting robots utilize the gripper that comes standard in the robotic arm (Barth et al., 2016).

Two main harvesting procedures were identified across the literature: (1) grasping, where the end-effector grabs the produce and then pulls or twists to harvest, and (2) cutting, where the end-effector cuts the produce peduncle to harvest (Fig. 8 B) and grabs the produce. Generally, grasping is the most widely used approach as it is easier to obtain 3D coordinates of large produce and these coordinates do not require extreme accuracy. An end-effector position accuracy of ±0.5 cm is widely accepted as satisfactory to grasp ripe produce because end-effectors typically accommodate varying fruit sizes (Bac et al., 2014).

Unfortunately, harvesting produce by grasping is limited when there are many obstacles such as leaves or branches as the end-effector might get stuck if other parts of the crop are reached. Other crops can be damaged if harvested by grasping (e.g. sweet pepper, lettuce), and thus

Table 2
Motion Delivery and End-Effector Performance Indicators

Crop (Reference)	Motion Delivery Subsystem				End-effector Subsystem	Crop (Reference)	Motion Delivery Subsystem				End-effector Subsystem
	Anthropomorphic Architecture	Cartesian Architecture	DoF	Visual Servoing	Harvest Method		Anthropomorphic Architecture	Cartesian Architecture	DoF	Visual Servoing	Harvest Method
Apple (Gené-Mola et al., 2019)			NA		NA	**Grapes and Apples** (Fernández et al., 2014)			NA		NA
Apple (Onishi et al., 2019)	x		6		Grasping	**Guava** (Lin et al., 2019b)	x		6		Peduncle
Apple (Feng et al., 2019)			NA		NA	**Kiwi** (Massah et al., 2021)			NA		NA
Apple (Yan et al., 2021)			NA		NA	**Kiwi** (Williams et al., 2020)	x		?		Grasping
Citrus (Lin et al., 2019a)			NA		NA	**Lettuce** (Birrell et al., 2020)	x		6		Peduncle
Citrus (Gan et al., 2018)			NA		NA	**Litchi** (C. Wang et al., 2017)			NA		NA
Citrus (C.H. Yang et al., 2020)			NA		NA	**Orange, pineapple, cucumber** (N. Guo et al., 2020)			NA		NA
Cucumber (Mao et al., 2020)			NA		NA	**Passion Fruit** (Tu et al., 2020)			NA		NA
Cucumber (Fernandez et al., 2018)			NA		NA	**Peach** (Wu et al., 2020)			NA		NA
Eggplant (Sepulveda et al., 2020)	x		6		Peduncle	**Strawberry** (Xiong et al., 2019a)		x	3		Peduncle
Eggplant, apple, grape (T. Zhang et al., 2020)	x		7		Peduncle	**Strawberry** (Zhou et al., 2020)			NA		NA
Eggplant, guava (Lin et al., 2020b)			NA		NA	**Sweet Pepper** (Arad et al., 2020)	x		6	x	Peduncle
Grapes (Kalampokas et al., 2021)			NA		NA	**Sweet Pepper** (Chen et al., 2020)			NA		NA
Grapes (Luo et al., 2016)			NA		NA	**Tomato** (L.L. Wang et al., 2017)	x		5	x	?
Grapes (Luo et al., 2018)			NA		NA	**Tomato** (Liu et al., 2019)			NA		NA

require that harvesting is performed by cutting the peduncle (Bac et al., 2016; Birrell et al., 2020). As such, these systems require the 3D location and orientation (3D pose) of the peduncle (cutting point). This is challenging as often the peduncle is smaller compared to the overall size of the produce (See Fig. 1) and is occluded by leaves or branches. Which can create multiple shadows, decreasing the produce detection accuracy of some detection methodologies; particularly for the produce of the same color as parts of the crop such as green sweet pepper. Increasing the localization difficulty. Another challenge is that produce does not normally grow perpendicular to the ground. It often grows with small-angle variations in all three axes that the end-effector needs to consider. As such, end-effectors that harvest produce by cutting the peduncle, often use manipulators with more degrees of freedom. Then again, having pose information has allowed the harvesting robot's success rate to significantly increase, and reduced the number of harvesting re-attempts (Kang et al., 2020). Effectively reducing the harvesting cycle.

2.4. Performance indicators

Because of the different harvesting robot subsystem architectures, researchers have developed multiple performance indicators that measure the robot's harvesting capabilities (See Table 1 and Table 2). This work uses a collection of categorical and continuous indicators to analyze the harvesting robot's vision subsystem performance, and where information was available, characteristics of the motion delivery,

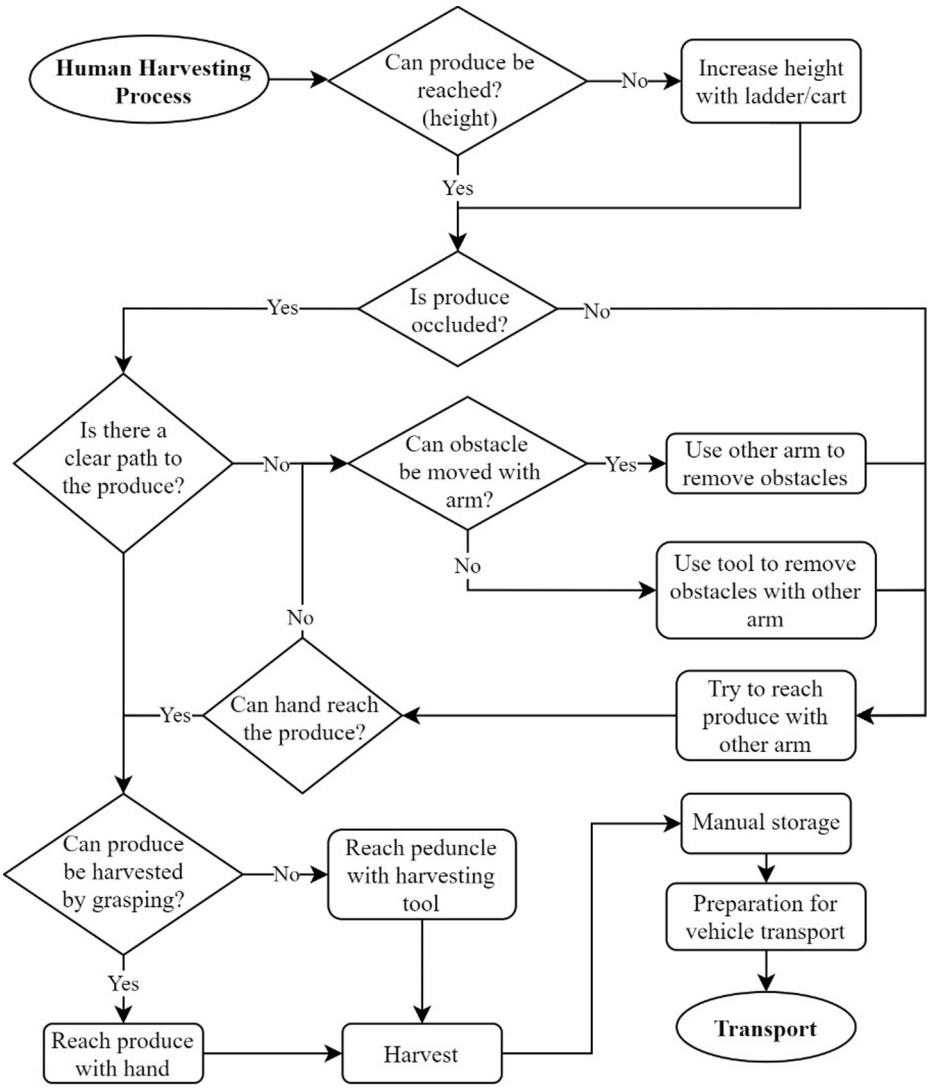

Fig. 7. Manual produce harvesting process.

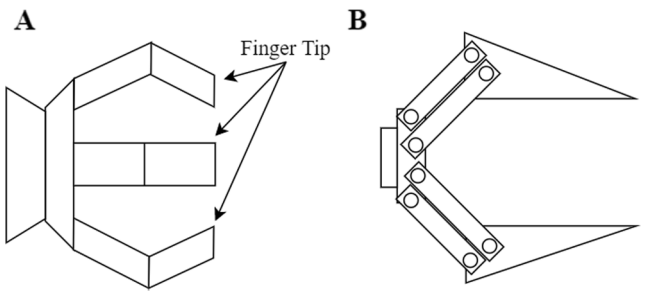

Fig. 8. (A). 3 Finger end-effector. Commonly used for harvesting produce of varying sizes by grasping. (B). 2 Blades end-effector. Commonly used for harvesting produce by cutting the peduncle.

and end-effector subsystems. The indicators included in this study were selected based on adoption across the literature.

For the vision subsystem, two categorical performance indicators were included: the environment where the project was tested (orchard, lab, greenhouse, field, and dataset) and whether the vision subsystem was capable of computing pose estimation for the produce (true/false). The difference between the environments is important as orchards and fields are usually much more unstructured than a laboratory or a greenhouse. Other variables such as lighting, obstacles, and the movement of the canopy due to wind can be controlled indoors. The distinction on whether the robot can provide produce pose information is important, particularly for produce that has a non-regular shape or produce that is required to be harvested by cutting the peduncle.

In addition, regarding the vision subsystem hardware, the sensor type, and computer's characteristics where the detection and/or localization algorithms are running are provided. For the hardware configuration, the available information is the CPU model and its maximum clock speed as well as the GPU model and its memory capacity. Furthermore, a brief description of the methodologies used for detection (Section 4) and localization processes (Section 5) is also present. An in-depth description of the vision sensor and the computer's hardware characteristics is present in Sections 3 and 4.5 respectively.

The following eight numerical performance indicators were analyzed for the vision subsystem:

1. Produce detection precision [%]: Is the ratio of correctly identified produce to the total number of produce (true positives and false positives) (Fernandez et al., 2018; Bac et al., 2014). Calculated as follows:

$$Precision = \frac{tp}{tp + fp} \quad (1)$$

where *tp* is the true positives or correctly identified produce and *fp* is the total number of false positives of incorrectly identified produce.

2. Produce detection recall [%]: Is the proportion of detected produce that is actual produce. Calculated as follows:

$$Recall = \frac{tp}{tp + fn} \tag{2}$$

where *tp* is the true positives or correctly identified produce, *fp* is the total number of false positives and *fn* the number of false negatives.

3. Produce detection F1 score [%]: Is the weighted harmonic mean of the recall and precision. When the recall and precision are given the same weight, it is calculated as follows:

$$F1 = 2\frac{precision \;\; recall}{precision + recall} \tag{3}$$

4. Produce detection time [s]: Is the total amount of time in seconds that a vision subsystem takes to process one image and detect all available produce.
5. Produce localization success [%]: It is the ratio of correctly localized produce to the total number of available produce (Ling et al., 2019). As mentioned, the position accuracy of ± 0.5 cm is widely accepted as satisfactory to grasp ripe produce because end-effectors typically accommodate varying fruit sizes (Bac et al., 2014).
6. Produce localization time [s]: Is the total amount of time that a vision subsystem took to compute the 3D coordinates of detected produce.
7. Harvesting success [%]: It is the ratio of successfully harvested produce to the total number of available ripe produce at the scene.
8. Cycle time [s]: The cycle time is the time the entire harvesting robot process took to harvest an individual fruit. Most researchers consider the cycle time (the lower the better) to be a key indicator in whether the research study can be transformed into a commercial product. However, this measure varies significantly from crop to crop (Tang et al., 2020) and requires field testing of a complete harvesting robot system. Which sometimes is not available as researchers focused on a specific subsystem. In Table 1, only 9 out of 30 projects were complete harvesting robot projects. The rest were projects focused on evaluating the performance of the vision subsystem.

For the motion delivery subsystem, two indicators are provided: (1) The motion delivery subsystem architecture (anthropomorphic/cartesian), and (2) the number of DoF that the motion delivery subsystem is using.

Finally, a categorical indicator was included for the end-effector subsystem: the harvest method. Which states whether the robot harvested produce by grasping or by cutting the peduncle. This distinction was included as produce that is harvested by cutting the peduncle is usually more difficult and could explain the decrease of performance for the research project.

3. Sensors for vision subsystems of harvesting robots

As discussed previously, the main components of a harvesting robot vision subsystem are a vision sensor and an onboard computer with detection and localization algorithms. While these components can vary significantly at each harvesting robot, the most significant differences found were the type of sensor that is used, and the detection and localization algorithms.

This section discusses the most common vision sensors in use at state-of-the-art harvesting robots (See Table 1). It first introduces 2D-capable sensors, analyzing their advantages, limitations, and use cases; finally, 3D-capable vision systems are introduced. A brief comparison of the vision systems discussed in this section is included in Table 3. Detection and localization methodologies will be discussed in the following sections.

3.1. 2D-capable sensors

These sensors provide two-dimensional data (2D). As they lack depth (3D) information, they are mostly used to identify the produce in a scene. Camera-based sensors are the most popular 2D-capable vision system (Table 1). This is because multiple single morphological features (such as color, shape, and texture) can be extracted from 2D images (Zhao et al., 2016). These features can then be used to identify produce and ripeness from a scene with different algorithms (Section 4). Limitations of camera-based 2D vision systems are that they tend to have difficulties while performing in highly occluded scenarios with varying lighting conditions (Lin et al., 2020b). Therefore, researchers have experimented with other 2D-capable sensors that are less sensitive to these conditions, such as hyperspectral, spectral, and thermal cameras. All of which will be discussed in this section.

3.1.1. Monocular camera

While multiple 2D-capable vision systems exist, most harvesting robots rely on the information of a monocular camera as the sensor used for the produce detection algorithm. The reason being that monocular vision systems have the lowest cost and their information can be used to identify fruits by shape, color, and texture (Zhao et al., 2016). Which can then be analyzed by multiple techniques, such as single feature and feature fusion analysis, supervised, and unsupervised learning (Garcia-Lamont et al., 2018). In this sensor, the effectiveness of image processing is due to the use of efficient image analysis algorithms, rather than the image quality (Mavridou et al., 2019). The most common sensors for this vision system are Charge Coupled Device (CCD) and Complementary Metal Oxide Semiconductor (CMOS) (Zhao et al., 2016).

Early fruit detection studies relied on the information produced by a black and white (B/W) camera. These cameras produced 2D grayscale images which were then processed for fruit detection via a geometric feature analysis (Whittaker et al., 1987). The absence of color information is the major disadvantage of B/W cameras. Making it difficult to achieve the desired level of accuracy using limited information from this sensor (Gongal et al., 2015). As color is one of the most important features in a crop, most recent harvesting robots use color cameras instead of W/B cameras for produce detection.

Color cameras are the most common because of their low price and the multitude of information contained in their images (Lee et al., 2010). This sensor allows the use of color-based segmentation, geometric and texture methodologies for produce detection (Gongal et al., 2015). Lin et al. (2020a) use color images to detect multiple fruits by using shape matching algorithms. Xiong et al. (2019b) use a color camera and a simple thresholding algorithm to detect strawberries. L.L. Wang et al. (2017) use color images of tomatoes to produce B/W segmentation by using color thresholds.

It is also possible to obtain 3D depth information with some errors from this sensor, however, most robots use a second 3D-capable sensor to improve localization results (Baeten et al., 2008). Feng et al. (2018) use a color camera and a laser for cherry tomatoes detection and localization.

Most color cameras provide information in the RGB color space; however, this space is inadequate for image processing. As there is a high correlation between the RGB components and the intensity is not decoupled from the chromaticity (Garcia-Lamont et al., 2018). This is why researchers perform image analysis in other color spaces such as HSV (Wu et al., 2020), HIS, L*a*b, and others (Zhao et al., 2016). Feng et al. (2018) use an R-G color model which intensifies the difference between the target fruit and the background.

Unfortunately, single color camera systems also have their limitations: they are very sensitive to lighting conditions and are not able to provide direct spatial location as they only provide 2D information. This

Table 3
Vision sensor features, localization principle, advantages, and limitations.

Vision Sensor		Available Detection Features	Localization Principle	Advantages	Limitations
Monocular Camera		Color, shape, and/or texture.	Optical relationship	Simplest Lowest cost	Variable lighting. Occluded environments.
Spectral Camera		The reflectance of the non-visible spectrum.	NA	Excels at similar colored scenes with variable lighting conditions.	High cost. Processing is time-consuming.
Thermal Camera		Temperature signature.	NA	Performs well with fruit with significant temperature variation from surroundings.	High cost. Produce with a similar thermic response as surroundings. Objects with different sunlight exposure.
LiDAR	2D	Backscattered energy.	ToF	High accuracy, scanning frequency, own source of light. Long Distance.	Requires additional movement for 3D information. High Cost.
	3D	Backscattered energy.	ToF	High accuracy, scanning frequency, own source of light. Long Distance. Direct 3D information.	High Cost. Low 3D spatial resolution. Computationally expensive.
RGB-D Camera	Stereo	Color, 3D shape, and/or texture.	Image Triangulation	Simplest 3D-capable sensor. Outputs 2D colored image and 3D data.	Computationally expensive. Same limitations as monocular cameras.
	Active	Color, 3D shape, and/or texture.	ToF	Own light source. Does not require postprocessing for 3D reconstruction.	Sensitive to strong lighting.

makes fruit recognition particularly difficult for low contrast scenes and images with similar colored backgrounds (Lin et al., 2020b).

3.1.2. Spectral cameras

While some objects might have a similar color in the visible spectrum, they might exhibit different reflectance in the non-visible spectrum (Feng et al., 2019). Making them easy to differentiate with spectral cameras. These sensors integrate standard color and spectroscopic imaging techniques into one system. There are two main spectral technologies: hyperspectral and multispectral. The main differences between the two, are the number of wavelengths, spectral resolution, and the way of acquiring data. Generally, multispectral cameras capture fewer than 10 discrete wavelengths. While hyperspectral capture more than 10 contiguous wavelengths (Lu et al., 2017). Most spectral cameras share three major components: (1) the light source which generates light to illuminate the target. (2) a wavelength dispersive device used to disperse the incident broadband light into different wavelengths. And (3), an area array detector. Used to measure the intensity of the collected light (Lu et al., 2017).

Feng et al. (2019) use a Forward Looking Infrared (FLIR) camera based on Multi-spectral Dynamic Imaging (MSX) technology for apple recognition. The authors state that this system could detect fruit with high precision with variable lighting and similar colored background scenes.

Unfortunately, it is difficult to use spectral imaging as this technology is limited by time-consuming data acquisition and analysis (Gongal et al., 2015). Therefore, hyperspectral imaging is difficult to implement in an online system. Instead, a few characteristic wavelengths are selected to speed up the process, which requires that the characteristic wavelengths be determined before using a system like this (H. Zhang et al., 2020). Spectral cameras also require more expensive devices when compared to a monocular color camera. Making them not affordable for practical use (Mavridou et al., 2019).

3.1.3. Thermal cameras

Closely related to spectral reflectance is the thermal response of objects (Kapach et al., 2012). This sensor captures the temperature signature of the objects (Gongal et al., 2015). Fruit detection is based on the assumption that their temperature varies significantly from their surroundings: as produce's mass is likely larger than leaves, their thermodynamic response should also be different (Lee et al., 2010). Gan et al. (2020) sprayed water in a citrus field forcing temperature changes in the fruit and the leaf surfaces. As this occurred at different rates, they were able to use a thermal camera to accurately detect citruses.

Unfortunately, detection problems can occur when the produce and the canopy have the same thermic response and different exposure to direct sunlight (Kapach et al., 2012; Gongal et al., 2015). This is why thermal cameras are often used along with other vision systems. Bulanon et al. (2009) and Gan et al. (2018) fused thermal and regular images to improve citrus detection.

3.2. 3D-capable sensors

As discussed earlier, 2D-capable vision systems are mostly used in produce detection. In contrast, 3D-capable sensors provide 3D coordinate maps of the scene which can be used to obtain the shape and spatial location of an object (Zhao et al., 2016). Therefore, these vision systems are mostly used in produce localization, working together with a 2D-capable sensor for produce detection. However, recently, 3D information is being used to detect produce by using 3D shape matching algorithms. This has the added benefit that both, detection, and localization can be performed with a single sensor.

3D information can be obtained with multiple sensors. The next section discusses the most popular used in current harvesting robots: starting with the less accurate monocular camera and moving to more accurate sensors such as 2D and 3D LiDAR, and RGB-D based cameras.

3.2.1. 3D monocular camera

As mentioned in the previous section, monocular cameras at most can only output 2D colored images. However, it is possible to obtain 3D information from a 2D-capable camera by knowing the relationship between the focal length of the camera and the pixel size; and the 2D location of the produce on a 2D image (Gongal et al., 2015). This enables 3D distance calculation from the produce to a 2D-capable monocular camera.

Baeten et al. (2008) utilize this technique to harvest apples with an anthropomorphic robotic arm mounted to a tractor. Their system uses two images captured at different distances of the same apple. This allows the system to calculate the distance and rotation angles of the end-effector to the fruit based on the apple's size. Mehta et al. (2017) use the same principle in a system that uses $\geq$2 inexpensive monocular cameras at different locations to perform estimation-based localization of oranges.

A major limitation of this technique is that as depth estimation is

based on similar-sized fruits, it will produce errors when applied at fruits of varying sizes. This technique also requires a moving sensor which can reduce the speed at which the manipulator can be moved towards the desired produce (Gongal et al., 2015).

3.2.2. LiDAR

Light Detection and Ranging (LiDAR) is a range-based method used to obtain distance information of objects. This is typically obtained using the time-of-flight (ToF) measurement: the time taken for a signal to reflect from an object (Gongal et al., 2015).

There are multiple LiDAR architectures: optomechanical, electromechanical, micro-electromechanical systems (MEMS), and solid-state. However, they all have four major subsystems: laser rangefinder, beam deflection, power management, and master controller units. What makes the architectures different is how their beam deflection mechanism works. Optomechanical LiDARs use optical components to deflect the beam, electromechanical sensors use electric motors and a mechanical stage, MEMS are micro-scaled versions of optomechanical LiDARs, and solid-state sensors use multiple stacks of 1-dimension LiDAR in one package (Raj et al., 2020). This sensor provides accurate distance measurements and the amount of energy backscattered from the object (Gené-Mola et al., 2019).

Researchers commonly use two types of LiDARs to produce 3D scene information: 2D and 3D LiDARs. 2D sensors are used for autonomous navigation by positioning a 2D LiDAR at the front of a robot, allowing the robot to detect obstacles (Le et al., 2020; Bonadies & Gadsden, 2019). As discussed in the previous section, it is also possible to use a 2D LiDAR to generate 3D data. This is accomplished by mounting the sensor on a slider or by moving the sensor while it generates 2D information; adding the third dimension (Eizentals & Oka, 2016; Bonadies & Gadsden, 2019). The advantages of these sensors include high scanning frequency, own source of light, and long-distance measurements. Some 2D LiDARs are also able to provide a 360° field of view (Chakraborty et al., 2019).

Eizentals and Oka (2016) use a 2D LiDAR mounted on a slider to generate point clouds (3D data) from the 2D slices of the sensor. This information is then used to locate the stems of green peppers in a crop, allowing the robot to compute the size and pose of the fruit. A major limitation is that this addition of a third dimension requires additional computation and time, potentially slowing the robot down.

Recent advances in this technology have made portable 3D LiDAR sensors commercially available. These sensors generate accurate direct 3D information of a scene without the need for manual implementation of a slider such as the one used by Eizentals and Oka (2016) as well as the respective transformations to obtain 3D information. The major advantages of these sensors are that the generated information is not affected by lighting conditions, and as the sensor provides the amount of backscattered energy, it is possible to detect and localize produce with this sensor (Gené-Mola et al., 2019). 3D LiDARs are currently more popular than 2D lidar and slider combinations for produce detection and localization.

Tsoulias et al. (2020) use a 3D LiDAR to detect apples in an orchard based on backscattered reflectance intensity and geometric features. Gené-Mola et al. (2019) use a reflectance threshold and 3D clustering algorithm to identify fuji apples with results comparable to RGB detection. With the added benefit that their method is not susceptible to variable lighting.

The major limitations of 3D LiDARs are high cost, 3D model generation is computationally expensive, low spatial resolution (Gené-Mola et al., 2019; Raj et al., 2020). However, modern lidars use multiple laser/detector pairs to address low resolution (Chakraborty et al., 2019).

3.2.3. RGB-D camera

These vision systems combine some of the previously discussed sensors. Red, green, blue, and depth (RGB-D) sensors can provide a color image (as a monocular camera) and pixel depth information in real-time (as 3D LiDARs) (T. Zhang et al., 2020). The colored image can then be used in produce detection by using multiple detection methodologies (Section 4) and the depth information is used for localization (Section 5).

There are two main types of RGB-D cameras: (1) Binocular. Which produces 3D information based on optical geometry. And (2), active cameras. Which typically combines a range-based ToF method and a monocular color camera (Tang et al., 2020). Active RGB-D cameras are becoming the most popular vision system for current harvesting robots (See Table 1). 3D data acquisition and processing are faster when compared to other systems while providing good accuracy of 3D scenes (Gongal et al., 2015).

Both systems will be discussed in this section. Carefully analyzing their working principles, advantages, and limitations.

3.2.3.1. Binocular (stereovision systems). This vision system uses two monocular cameras separated by a known distance with an angle between them. This sensor uses a passive optical geometry-based method to produce 3D data (Tang et al., 2020). When used, both cameras capture a slightly different color image of the same scene. Then, depth information can be obtained with additional processing through triangulation between those two images (Zhao et al., 2016).

C. Wang et al. (2017) used this sensor to identify Litchi fruits in an orchard using supervised classifiers and clustering algorithms. The two main performance shortcomings reported were attributed to varying illumination and occluded scenes. Both of these are limitations of monocular cameras. Ling et al. (2019) used a binocular camera for a dual-arm tomato harvesting robot reporting high detection rates. However, scenes with high occlusion were also a limitation. Small localization errors can accumulate and produce inaccurate 3D scene reconstruction. This type of sensor also struggles to accurately calculate depth when presented with scenes that have symmetric patterns. Which under certain situations can happen when multiple produce is distributed across the scene.

Limitations of this sensor, as stated previously, are variable lighting and highly occluded scenarios; the generation of the 3D model is also computationally expensive and time-consuming (Gongal et al., 2015; Gené-Mola et al., 2019).

3.2.3.2. Active. Active 3D cameras are becoming the most used vision system in harvesting robots (See Table 1). This sensor provides a colored 2D image and accurate 3D spatial data from the scene (Lin et al., 2019a). This system consists of an infrared (IR) light source and an IR camera that produces 3D information, and a monocular colored camera. The IR camera can measure distances as it detects the phase shift that was received after it left the IR light source (ToF) (Lin et al., 2019b). These sensors can be faster when compared to stereovision systems (Gongal et al., 2015). As they also have their own source of light, they are reliable under low light situations (Tu et al., 2020).

Lin et al. (2019a) used an active RGB-D camera, a depth filter, a Bayes-classifier, and a clustering algorithm to detect and localize citrus in an orchard reporting millimeter range localization accuracy. A major limitation was that under strong lighting (sunlight) conditions, some clusters were incorrectly classified. Tu et al. (2020) used color and depth information to detect passion fruit. The authors reported that their color and depth detectors outperformed single color and single depth detectors.

A limitation of these systems is that they can give inaccurate measurements when under strong lighting conditions such as direct sunlight (Lin et al., 2019a; Tang et al., 2020). Because of this, researchers suggest using light shields and/or avoiding direct sunlight harvesting conditions to reduce the influence of light on this sensor (Wu et al., 2020; Lin et al., 2019b).

3.3. Multisensory systems

While there is a clear increase of projects using a single sensor for produce detection and localization such as RGB-D based one (11 out of 30 at Table 1), there are still many research projects that use multiple sensors to detect produce with the objective of increasing the detection performance and/or providing information for localization.

Bulanon et al. (2009) demonstrated that using color information along with thermal images, improved the detection of oranges. Gan et al. (2018) combined the information of a thermal and an RGB camera to obtain better performance at detecting citruses when compared with color-only analyses: reporting an increase of almost 9% in recall and precision when fusing information of both sensors.

Multisensory systems can also be used for different processes. Sepulveda et al. (2020) used an RGB camera and a ToF-based sensor to detect and estimate the pose of eggplants. Here, color information was used to detect eggplants, then this information was later used to segment the point cloud obtained from the ToF camera. The 3D information obtained from this sensor, also allowed the robot to perform pose estimation of the produce. As mentioned, this is required for produce that must be harvested by cutting the peduncle, such as eggplants. Fernández et al. (2014) used three sensors to evaluate their performance on identifying apples and grapes: a color camera, multispectral camera, and a ToF camera. The authors conclude that the main advantages of a system like this are that it can provide all the necessary data to detect and locate produce without performing any pre-treatment on the images before processing and any preparation of the crops

4. Detection methodologies

As mentioned before, harvesting robots are expected to work in multiple different environments with highly variable factors. As a result, researchers are constantly experimenting with combinations of different detection algorithms and features. The detection process is crucial for the harvesting success of the robot, as the next processes will often depend on the quality of the detection process outputs and the 3D information provided by the vision subsystem sensor (see Fig. 5).

This section first introduces the most widely used features that can be extracted from the sensor's data. It then introduces single and multiple feature analyses; methodologies that rely on the use of simple morphological features such as color, shape, texture, and image processing techniques. Then, machine learning-based classifiers are introduced: unsupervised, supervised, and neural networks. In other words, algorithms that can learn from data through an iterative training process (Zhao et al., 2016). Finally, a comparison of the requirements, advantages, and limitations of the current detection methodologies is presented in Table 4.

4.1. Feature extraction

Once data is obtained from the vision system, it first needs to be processed to identify if the target produce is present at the scene (detection), and then localize it (Section 5). As discussed before, harvesting robots are expected to perform in many different environments with multiple variations. Most of the encountered variations are attributed to crop, robot, and environment variables; and detection algorithms are directly influenced by these factors (See Fig. 4) (Arad et al., 2019). Before detecting the produce, it is necessary to extract features from the sensor's data with which the detection and localization algorithms will work.

The number of features available will often depend on the type of sensor that is being used and the data that it provides. The most widely used features for produce detection rely on morphological properties of the crop such as shape, color, and texture (Zhao et al., 2016). These features can be provided by 2D-capable sensors and work on the assumption that ripe produce characteristics vary from other parts of the

Table 4
Produce Detection Methodologies Advantages and Limitations

Detection Methodology	Dataset	GPU	Advantages	Limitations
Single Feature Analyses				
Color			Simple threshold algorithm. Inexpensive sensor.	Susceptible to varying lighting. Low performance with similar colored produce to other parts of the crop.
Geometric			More reliable under different lighting conditions.	Performance is limited in highly occluded scenarios. Limited to crops of simple geometrical shapes.
Texture			Reliable under different lighting conditions and occluded scenarios.	Only feasible on produce with very contrasting skin against other crop parts.
Spectral Reflectance			Very reliable under varying lighting and occluded scenarios.	Data acquisition is very time-consuming. Only feasible when produce has very different spectral reflectance against other crop parts. Requires expensive sensors.
Thermal Response			Very reliable under varying lighting and occluded scenarios.	Only feasible in outdoor production environments. Performance is limited when produce has the same temperature as other parts of the crop. Expensive sensor.
Multiple Feature Fusion			More reliable than single feature analyses. Often execute faster than ML techniques.	Requires multiple algorithms. Lower detection rates than ML techniques.
Unsupervised Learning			Reliable for techniques that require clustering. Often used to filter data.	These algorithms are almost always combined with supervised learning.
Supervised Learning				
Bayesian	X		Fast execution.	Requires probability information from images. Often not available.
KNN	X		Useful for color-matching techniques.	Requires large storage for training samples and is computationally expensive.
SVM	X		Can classify non-linearly separable data. Computationally efficient.	Offers lower performance when under very different lighting conditions and/or object visibility is reduced.
Deep Learning				
R-CNN and Variants	X	X	Better accuracy than SSD and YOLO.	Require large, labeled datasets.
SSD	X	X	Faster performance than R-CNN.	Long training time. Require powerful hardware.
MS-FRCN	X	X	Better performance for small objects.	

(*continued on next page*)

Table 4 (*continued*)

Detection Methodology	Dataset	GPU	Advantages	Limitations
YOLO	X	X	Faster than SSD and R-CNN, Offers similar performance to SSD.	Poor performance on small close-together objects. Requires large, labeled datasets. Long training time. Requires powerful hardware. Not as accurate as R-CNN.

crop (leaves and crops). Unfortunately, morphological features have their limitations when the sensor is presented with variable lighting or strong occlusions.

Color is greatly affected by varying light conditions as its tone will change. While 2D geometry and texture are not greatly affected by lighting, they are mostly used when produce has simple geometrical shapes (e.g. apples and tomatoes can be modeled as a circle in 2D and as a sphere in 3D) or very different skin textures from other parts of the crop (e.g. pineapples) (Lin et al., 2020a). Unfortunately, 2D geometric feature extraction can be difficult as it is very sensitive to occlusions from objects (leaves and branches) (Gongal et al., 2015; Chaivivatrakul & Dailey, 2014).

Researchers have also experimented with features that are less susceptible to lighting conditions such as spectral reflectance and thermal response. Unfortunately, these features require specific vision sensors: spectral and thermal cameras. As a simple monocular sensor will only be able to provide information from the visible spectrum.

Spectral reflectance is the reflectance of produce in the non-visible spectrum. This feature is used in produce that might have a similar color in the visible spectrum to other parts of the crop, but different reflectance in the non-visible spectrum. In contrast to morphological features, spectral reflectance is less susceptible to light variations as the vision system has its own light source (Feng et al., 2019; Lu et al., 2017). However, the main limitation of using spectral reflectance is that it has very time-consuming acquisition and processing (Gongal et al., 2015).

A thermal response analyzes the temperature signature of the produce (Gongal et al., 2015). This feature is based on the assumption that the produce temperature varies significantly from their surroundings (Lee et al., 2010). Unfortunately, this feature will be limited when the crop's canopy has a similar thermal response to the produce (Kapach et al., 2012).

With the recent availability of affordable 3D-capable sensors, current research is focusing on extracting features from 3D information. Assuming that 3D data is available, the surface normal and curvature can be indirectly calculated (Wu et al., 2020). The advantages of these features are that they are not easily affected by varying lighting conditions or occluded scenarios.

As stated, these features can then be used in single or multiple feature analyses, and machine learning classifiers.

4.2. Single feature analysis

Single feature analyses consist of mostly analyzing morphological features such as color and shape (See Fig. 9) by using common image processing techniques. Such as color threshold, simple shape-matching algorithms, and morphological operations for image processing. As only one feature is used, this technique is often the simplest and therefore, is very susceptible to external variations (varying lighting and highly occluded scenes). Because of this, most harvesting robot vision systems research has focused on implementing artificial intelligence methodologies. However, while single feature analysis techniques' accuracy might vary in difficult scenes (varying lighting, multiple occlusions, similar colored background, etc.), they require the lowest amount of computing power and are some of the simplest to implement. Because of these characteristics, there are cases where these techniques might be a better fit. Especially where hardware availability such as a GPU is limited.

Fig. 9. Crop morphological feature variance. (A) Tomatoes and (C) red peppers are suitable for color, shape, and texture detection. (B) Green pepper (red outline) for shape and texture detection. (D) Color detection can be applied for cucumber (in red); however, shape or texture detection might provide better results. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

While current harvesting robot vision subsystem research is focusing on applying Artificial intelligence techniques such as deep learning, there are still multiple research projects that rely on single and multiple feature analysis. This section provides a brief overview of recent work that involves the analysis of simple morphological features that have competitive performance, even to machine learning techniques.

4.2.1. Morphological features

Perhaps the simplest and earliest morphological feature analysis for produce detection is the use of color. In a color-based segmentation algorithm for fruit recognition, the image pixels are categorized into whether it belongs to the target fruit or the background by a color threshold (See Fig. 9 A and C) (Zhao et al., 2016). Color analyses, are typically done in the HSV color space (Xiong et al., 2019a; Wu et al., 2020; Eizentals & Oka, 2016). As it has a component for pixel brightness that is not easily affected by color depth as the RGB color space. Researchers have also used other color spaces such as HIS, L*a*b*, YIQ, OHTA, and LCD that are also less susceptible to the influence of varying illumination (Zhao et al., 2016). Unfortunately, these methods are limited and may fail to segment overlapping fruits of similar color into individual ones in occluded scenarios and are very sensitive to light variations (L.L. Wang et al., 2017; Lin et al., 2020b).

L.L. Wang et al. (2017) were able to detect tomatoes by first normalizing the light intensity of red and green color components. Then, the Otsu algorithm is used to segment the image pixels into two classes by a threshold. While the detection results are promising, the harvesting robot had a maximum harvesting rate of 88%. Xiong et al. (2019b) use the RGB information of an RGBD-infrared camera for a simple color threshold algorithm that detects strawberries. Then, simple morphological operations (erosion and dilation) are used to retrieve binary images.

To negate the influence of varying illumination and to ease the detection of produce with a similar color to the background, researchers have opted to use algorithms based on geometric feature extraction (Gongal et al., 2015). This approach relies on the use of simple shape models where fruits can be reasonably modeled, such as circles in 2D and spheres in 3D (Zhao et al., 2016; Kapach et al., 2012).

Lin et al. (2020a) used a partial shape matching algorithm to detect the contour of multiple fruits from colored images. This algorithm aims to detect candidate sub-fragments that are similar to a reference contour (shape descriptor). Lin et al. (2020b) use a 3D shape matching algorithm to detect green peppers and eggplants from a point cloud. Tsoulias et al. (2020) used the curvature and reflectance as a threshold to segment a point cloud by differentiating between leaves, branches, stems, and apples. Li et al. (2018) used the normals from a point cloud to calculate the symmetry axis of sweet pepper. The authors state that this allows for more accurate pose detection regardless of whether the peduncle is small or occluded.

Assuming 3D information is available, other techniques such as 3D clustering and 3D shape matching can be applied. 3D clustering algorithms utilize a known distance measurement to separate data into clusters (Kapach et al., 2012). When processing a point cloud, this method allows performing fruit separation via 3D clusters (Gené-Mola et al., 2019). Unfortunately processing large amounts of 3D information can be slow.

Wu et al. (2020) used Euclidean clustering to improve the results of an SVM (Support Vector Machine) classifier to detect peaches in a point cloud. Tsoulias et al. (2020) used a density-based scan algorithm (DBSCAN) to cluster previously segmented points to find individual apples. Lin et al. (2019a) used this technique to process RGB-D images from an active camera. As the resulting clusters could represent a citrus, a leaf, or some fruit, the clusters were also processed by an SVM classifier to exclude non-fruit clusters.

3D shape matching works by fitting a 3D model into any cluster available at the point cloud. However, because of the low resolution of 3D sensors, this method is mostly used for simple regular-shaped produce; as they can be modeled with simple geometrical figures such as a sphere or a cylinder (Lin et al., 2020b). The main advantage of this method is that by fitting a model in a 3D space, in addition to the position, the object's orientation will also be available.

Ge et al. (2020) used a shape completion algorithm for symmetrical fruits (strawberry, apple, oranges, etc.). The algorithm reconstructed the produce point cloud based on symmetry planes. This procedure was tested to localize strawberries reporting high accuracy. Lin et al. (2020b) used a 3D shape detection algorithm to detect fruit clusters in a point cloud. The advantage of this method is that multiple shapes can be used. In this case, a sphere was used to detect pepper and guava, while a cylinder was used to detect eggplants.

Limitations of this method are that shape can be difficult to extract as it is time-consuming, computationally expensive, and is very sensitive to occlusions such as leaves and branches (Kapach et al., 2012; Gongal et al., 2015). As the occlusions will change the resulting shape, its size, and other geometric attributes.

Texture might be one of the first visual cues that allowed to completely describe an object as it is a pattern that repeats itself. This property allows for the discrimination of different object types: the skin of the fruit can be distinguished from other parts of the plant (leaves and branches) by using edge detection (Kapach et al., 2012). One of the benefits of this analysis is that the surface color does not have an impact on texture. Therefore, texture analysis can be used for fruit with a similar color to other parts of the crop (Gongal et al., 2015). It is common to combine this technique with additional visual cues as it can also be affected by occlusion (Kapach et al., 2012).

Kadir et al. (2015) used a texture-based analysis combined with a shape detection algorithm for mango detection in scenes with complex backgrounds. Chaivivatrakul and Dailey (2014) used local texture descriptors to identify pineapple and bitter melon in the field. Reporting that this method was very accurate for textured fruit.

Unfortunately, this method is also sensitive to variable lighting and has only proofed accurate with fruits of rough texture (such as pineapples) (Gongal et al., 2015). Fruit with a smoother texture will introduce errors in this analysis (Chaivivatrakul & Dailey, 2014).

4.2.2. Spectral reflectance

Spectral reflectance analysis is exclusive to hyperspectral or multispectral cameras. This technique analyzes the reflectance of produce in the non-visible spectrum via a threshold, whose values are often different even if the produce has a similar color in the visible spectrum (Feng et al., 2019). As mentioned previously, it is common for these cameras to have their own light source as a component. Because of this, the analysis is less susceptible to light variations.

Feng et al. (2019) used a threshold algorithm that processed the spectral reflectance to accurately detect apples in an orchard. Okamoto and Lee (2009) used noise reduction filtering, labeling, and thresholding to detect citrus using hyperspectral imaging.

Limitations of spectral reflectance are that data acquisition and processing are very time-consuming and objects with similar spectral reflectance will also introduce errors in this analysis (Gongal et al., 2015; Okamoto & Lee, 2009).

4.2.3. Thermal response

This analysis is exclusive to thermal cameras, as it requires dedicated hardware to obtain the thermal signature of an object. Like spectral reflectance, this technique analyzes the thermal response by using a simple threshold algorithm. This analysis is used in produce with a similar color to other parts of the crop.

Gan et al. (2018) fused the thermal response with color information to get better performance at detecting citruses in a field when compared to color-only analyses. Bulanon et al. (2009) used a similar approach to fuse thermal response and color information to improve oranges detection performance.

As discussed, even though it is possible to perform thermal response-only analysis, recent projects combine thermal information with other features such as colored images to improve produce detection (Gan et al., 2020; Bulanon et al., 2009; Gan et al., 2018).

4.3. Multiple feature fusion

When testing under uneven or uncontrollable lighting, similar colored backgrounds to the target fruit, and occluded surfaces, researchers found that by using multiple features ($\geq$2), the overall recognition reliability is improved (Zhao et al., 2016). This is because features have different limitations: color detection is very sensitive to light changes; however, shape and texture are not. Yet, shape and texture are greatly affected in occluded and overlapping scenarios, where the color might not. Fusing multiple features is the standard of harvesting robots that do not use ML techniques. As researchers use multiple combinations to further improve the accuracy of their algorithms.

Arad et al (Arad et al., 2019), used a hue and saturation threshold to remove most of the background of the original image. Then, they use a minimum object size shape filter to create a cleaner image where sweet pepper can be later classified for ripeness. Here (Gené-Mola et al., 2019), researchers use a reflectance threshold and a 3D shape matching to detect apples.

While simpler multiple feature algorithms such as threshold or shape detection can provide accurate results, there is a clear trend for analyzing multiple features using machine learning algorithms. As performance is often improved. These algorithms are analyzed in the following sections.

4.4. Machine learning classifiers

Machine learning is a branch of computer science and artificial intelligence that designs and analyses algorithms that improve their performance based on observable data (Kapach et al., 2012). Machine learning algorithms learn from the data through an iterative training process that allows them to improve after each iteration (Zhao et al., 2016). In recent years, the use of ML classifiers for produce detection in harvesting robots has grown substantially (see Table 1). Gongal et al. 2015) concluded that this is because they have better performance than single feature analyses especially when lighting varies significantly.

While the focus of this work is on discussing the recent advances of the vision subsystem of harvesting robots, ML algorithms have many other applications in the agricultural industry. Drury et al. (2017) concluded that because Bayesian networks can work with incomplete information while also incorporating new data, they are suitable for automated monitoring, predicting, identifying causes, classifying data, and even supporting decision making. Roy and De (2020) demonstrated that genetic algorithms can be successfully used to predict the weather, and as such determine if watering is needed.

Harvesting robots use two main types of ML algorithms: supervised, which requires a labeled dataset, and unsupervised. Which does not. An area of recent major development in harvesting robots is the use of deep learning techniques. This area aims to further improve the produce detection accuracy by adding more complexity to the ML models (Kamilaris & Prenafeta-Boldú, 2018). Because of the multiple research projects found in the literature, this technique is presented as a separate section below. This section first introduces the most common unsupervised and supervised techniques. It then introduces deep learning, and its most common techniques used in harvesting robots nowadays.

4.4.1. Unsupervised

As mentioned before, unsupervised classifiers do not require a labeled dataset (Rehman et al., 2019). Instead, unsupervised algorithms separate the data into different clusters based on a distance measurement applied to the database features (Kapach et al., 2012). In other words, unsupervised learning algorithms are clustering algorithms, where the main difference relies on how the distance metric is calculated.

Many algorithms can be used for unsupervised clustering, however, in harvesting robots, one of the most widely used is K-means clustering. K-means clustering principle is to find the K number of clusters in a given dataset. The algorithm works iteratively by assigning each data point to one of the K groups based on the multiple features provided. Then, the data points are clustered based on feature similarity (Rehman et al., 2019). This algorithm minimizes the sum of the distance between objects and their respective clusters centers (Gongal et al., 2015).

Luo et al. (2018) used K-means clustering to segment an image based on a color component to identify grapes in a vineyard. This image was further processed to remove noise by using edge detection. Méndez et al. (2019) use 3D data from a LiDAR and K-means to detect oranges by shape.

However, these algorithms rely on data that can be easily clustered, which is not always the case. Because of this, most of the vision systems analyzed complement unsupervised clustering algorithms with supervised algorithms.

4.4.2. Supervised

Contrary to unsupervised classifiers, supervised classifiers require a class-labeled dataset. These classifiers generally provide better performance than single feature-based and unsupervised methods (Gongal et al., 2015). However, as mentioned, they require an additional step before being implemented: training. To train a supervised algorithm, a dataset of features labeled with classes is required. Unfortunately, this is a manual tedious procedure for large datasets that is prone to user bias when labeling ground truth data (Koirala et al., 2019). These learning methods also require greater computational resources and large amounts of labeled data when compared to single feature analyses. Fortunately, high-performance graphics processing units (GPUs) have become available and labeling has gotten easier with the use of freely available graphical annotation tools (Gongal et al., 2015; Koirala et al., 2019).

These classifiers offer more precise detection rates as researchers can use multiple features to train the algorithms. There are many supervised classifiers that work in different ways. The most widely used by harvesting robots are discussed in this section.

4.4.2.1. Bayesian. One of the earliest supervised classifiers used for produce detection was the Bayesian classifier (Slaughter & Harrell, 1989). This is a probabilistic classifier based on Baye's theorem. It makes statistical interpretations that are based on prior knowledge and probability distributions (Gongal et al., 2015).

Slaughter and Harrell (1989) used a Bayesian classifier based on color to detect oranges. The authors attributed the limitations of the algorithm to the varying information from the scene, such as lighting conditions and occlusions.

Unfortunately, the main limitation of this classifier is that it needs prior probabilities information from training images that are not always available (Gongal et al., 2015).

4.4.2.2. KNN. K-nearest neighbors (KNN) is a supervised clustering learning method. KNN classifies an unknown feature vector to the class that most relates through common attributes within its K nearest neighbor (Gongal et al., 2015). Unknown features are classified by computing a similarity measure (Euclidean distance) to the training observations. As with most clustering algorithms, the similarity measure can be modified to others, such as Manhattan distance, Chebyshev, and Mahalanobis depending on the properties of the available data (Rehman et al., 2019).

Pourdarbani et al. (2019) used KNN to detect plums by classifying pixels by color into two classes: fruit and background. The authors concluded that KNN provides better detection results when provided with correctly labeled data than a single artificial neural network (ANN) heuristics.

The main limitations of KNN are that it requires the storage and processing of all the training samples during the classification process (Pourdarbani et al., 2019).

4.4.2.3. Support Vector Machine (SVM). One of the most widely used supervised classifiers is the Support Vector Machine (SVM). This method works by constructing an N-dimensional hyperplane that separates data into two classes optimally: maximizing the distance between the separating hyperplane to the nearest neighbor of each class (Kapach et al., 2012). As the data space is transformed into a higher-dimensional space where classes become linearly separable, SVM can also be used to classify data that is non-linearly separable (Rehman et al., 2019).

Pourdarbani et al. (2019) compared SVM with other supervised classifiers to detect plums by classifying pixels by color. The authors concluded that in contrast to KNN, SVM also achieves good results while being computationally efficient. As it only stores the selected support vectors. Liu et al. (2019) used SVM to detect tomatoes using colored images. Their method used data from histograms of oriented gradients (HOG) to train the SVM algorithm. This methodology reduces the influence of varying illumination and occlusion while still providing better performance than single-feature detection methods.

4.4.3. Deep learning

Deep learning (DL) is an extension of classical ML methods for object detection. It adds more complexity to the model and transforms the data using multiple functions that allow data representation through multiple levels of abstraction (Kamilaris & Prenafeta-Boldú, 2018; Koirala et al.,

2019). Deep learning combines high-level features with low-level features to produce distributed feature representations of the data (Tang et al., 2020). Currently, there is a lot of development in deep learning applications for harvesting robots. These techniques excel at solving complex problems well and fast (through parallelization) if provided with adequately large datasets describing the problem. While DL is popular in multiple applications dealing with raster-based data (videos and images), it can be used with any form of data (Kamilaris & Prenafeta-Boldú, 2018).

In contrast to most supervised classifiers, DL does not require manual feature extraction. It automatically performs feature selection and classification from any given labeled dataset. This is done, by using multiple layers of neural networks that produce a vector of distinct features (Koirala et al., 2019).

Convolutional Neural Networks (CNNs) constitute the basis of DL. CNNs are made by deep (multiple layers) feed-forward ANNs (Kamilaris & Prenafeta-Boldú, 2018). CNNs have been widely used for image processing such as object detection. As they can learn translational invariant patterns, they are able to accurately detect objects wherever they are placed on an image. A basic CNN consists of an input and an output layer, with intervening convolution and pooling/sub-sampling layers (Koirala et al., 2019). Early layers of a CNN learn simple local patterns (e.g., edges), while later layers capture more semantic representations of an object (e.g. shape). During training, the CNN convolution layers act as filters that extract useful information for each of the object's classes. This information is later used to perform object classification (Koirala et al., 2019). Early CNN for object detection used a sliding window classifier. Unfortunately, using all the information obtained from this window, slowed down object detection (Koirala et al., 2019). Fortunately, there has been a lot of development in deep learning. As such, current state-of-the-art harvesting robots utilize new DL architectures such as Region-CNN (R-CNN), Faster R-CNN, Single Shot MultiBox Detector (SSD) and You Only Look Once (YOLO). Which are faster.

There are two main types of DL CNN outputs (see Fig. 10). The most common are the locations of bounding boxes of varying sizes where the object was detected. The second type are masks of pixels where each color represents an object.

The disadvantages of DL are that it generally takes longer training time, problems might occur when using pre-trained models on small or significantly different datasets, there might be optimization problems due to the models' complexity, as well as hardware restrictions (Kamilaris & Prenafeta-Boldú, 2018). However, they consistently show the best detection performance rates even on very complex scenarios (varying lighting and highly occluded scenes).

Recent harvesting robots use many different deep learning architectures. This section introduces the most widely used while analyzing their differences and use-cases.

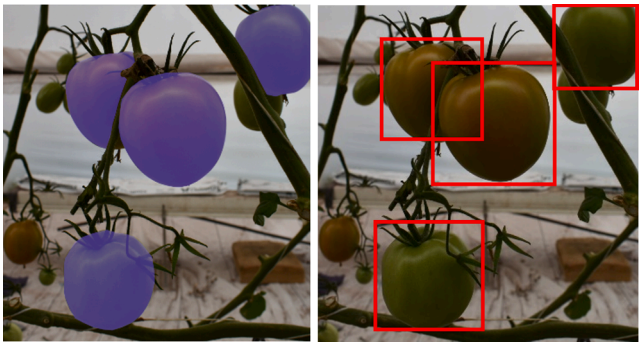

Fig. 10. DL CNN outputs by architecture. (Left) CNN segmentation. (Right) CNN Bounding box.

4.4.3.1. R-CNN, fast R-CNN, faster R-CNN, and mask R-CNN. Using a sliding window where the classifier is run at evenly spaced locations results in a slow object detection CNN. R-CNN replaced this method with a faster heuristic Selective Search algorithm which filters out some regions (Koirala et al., 2019). Fast and Faster R-CNN were developed to improve the detection speed of R-CNN. This is achieved by using a two-stage detection architecture, where the first stage is region proposal followed by a classification/detection stage (Koirala et al., 2019).

P. Lin et al. (2020) used R-CNN to detect strawberry flowers in an outdoor field. Hu et al. (2019) used Faster R-CNN to detect ripe tomatoes in a greenhouse. Gan et al. (2018) also used this architecture to first detect citrus fruit by color and then use thermal reflectance information to further improve the accuracy. Williams et al. (2020) used Faster R-CNN as a replacement of a semantic segmentation approach in a kiwi harvester robot. The authors argue that the selection was made as this architecture has been designed to operate in real-time while showing stat-of-the-art performance.

Mask R-CNN is an extension of Faster R-CNN for instance segmentation. This architecture outputs the location of exact pixels followed by masks for each object inside the bounding box (See Fig. 10; Koirala et al., 2019).

T. Zhang et al. (2020) used Mask-RCNN to process color images and detect multiple produce in a harvesting robot in an artificial indoor environment. C.H. Yang et al. (2020) also used this technique to identify branches and citrus fruits in an orchard. The authors state that branch information is important to reduce end-effector collisions while harvesting.

Currently, researchers are testing deep learning architectures with 3D information such as point cloud segmentation. In harvesting robots, the objective is to segment the 3D point cloud by obtaining a set of clusters that could represent produce (Gené-Mola et al., 2019).

Gené-Mola et al. (2019) used a reflectance threshold to segment apples in a point cloud obtained from a LiDAR. Then, the point cloud was processed using KNN to remove outlier points. Finally, DBSCAN (density-based scan algorithm) was used on the segmented pixels to create clusters of unique apples. Gené-Mola et al. (2020) used a Mask R-CNN to segment 2D colored images to detect apples. Then, a projection of the 2D image detections was made onto 3D space. Effectively segmenting the point cloud. Zeng et al. (2020) used a 3D LiDAR and clustering algorithms to segment a point cloud into trellis wires, support poles, and apple tree trunks.

4.4.3.2. SSD. For further speed improvement in object detection, SSD (Single Shot MultiBox Detector) and YOLO (next section) removed the region proposal stage found at Fast and Faster R-CNN. These CNNs were designed to consider the dense sampling of possible object locations for object detection. SSD simultaneously predicts the object class and bounding box, making it faster than Faster R-CNN (Koirala et al., 2019).

Onishi et al. (2019) used SSD to accurately detect apples in an orchard by using images from a stereo camera. Arad et al. (2019) used this architecture because of its high speed and accuracy to detect sweet pepper for a harvesting robot. This work incorporated an experimental Flash-no-Flash (FNF) illumination system which guaranteed constant illumination for all the produce while harvesting. (Gan et al., 2020) used thermal imaging information and SSD to detect immature citrus fruits.

4.4.3.3. MS-FRCNN. Multiple Scale faster region-based convolutional networks (MS-FRCNN) is an architecture used to improve the detection of small fruits. This architecture incorporates feature maps from shallower convolution feature maps for regions of interest pooling (Tu et al., 2020).

This architecture was used by Tu et al. (2020) to detect passion fruit by using colored images from an RGB-D-based camera. The authors stated that performance improved when depth information was combined with RGB data. This architecture also achieved better overall

performance when compared to other supervised classifiers (SVM) and DL architectures (Faster-RCNN).

4.4.3.4. YOLO. YOLO (You Only Look Once) converts the input image to a tensor of scores for object detection. This architecture is one of the fastest detection methods as it can predict the class and bounding box coordinates from the final feature map in a single forward pass through the CNN. There have been multiple iterations of this architecture that have improved the overall speed and accuracy. YOLOv3 released in 2018, achieves an accuracy similar to SSD at three to four times the speed (Koirala et al., 2019).

Birrell et al. (2020) selected YOLOv3 because of its speed at detecting objects to detect iceberg lettuce in a field. Liu et al. (2020) used the same architecture to detect tomatoes, arguing that its principal advantage is its detection speed while still maintaining reasonable accuracy.

The main limitation of YOLO architectures is that they offer poor performance on very small close-together objects (Birrell et al., 2020). This is because the architecture of YOLO subdivides the input image into smaller regions, which makes it harder to detect these objects.

4.5. Vision subsystems hardware characteristics

One limitation of using deep learning for produce detection is that powerful hardware is required to speed up both, the training, and the inference processes. And until recently, most research projects used the same computer for both processes. Which often increased the weight, size, and power consumption of the vision subsystem. In Table 1, the hardware where the produce detection and localization algorithms were implemented is available. While some projects do not specify any hardware (6), most included at least the model of the CPU that was used.

Of all the available configurations (24) the least powerful CPU for a project that did not use a deep learning model was an i3 4150 @ 3.5 GHz. For a project that used deep learning, the slowest CPU used was an i5 3210 M @ 3.5 GHz paired with an NVIDIA GTX 1060 6 GB graphics processing unit with a core speed of 1.4 – 1.67 GHz. As expected, research projects that involved the use of CNN used powerful CPUs and GPUs. However, projects that relied on other techniques such as color segmentation, reflectance thresholds, or classifiers that do not rely on CNN, still used mid to high-end CPUs. Fortunately, with the recent availability of Edge AI hardware accelerators projects that use a powerful computer to train a deep learning model and an Edge AI embedded computer for deployment is starting to be implemented (Kalampokas et al., 2021). This has the benefit that Edge AI accelerators are smaller, more efficient, without compromising on detection accuracy and real-time performance.

Table 5 presents a brief comparison of the hardware that was used by type of methodology and the average time the vision subsystem took for produce detection. From these projects, the least and most powerful hardware was then selected using the CPU as the criterion. Contrary to what was expected, algorithms that used DL executed the fastest (2.29 s), followed by methodologies that used a combination of deep learning and supervised/unsupervised algorithms (6.29 s), then, single/multiple feature analysis (5.26 s), and finally supervised/unsupervised methods (10.09). The main reason that single/multiple analysis did not outperform DL techniques, is that most hardware did not include a dedicated GPU; and where available, the algorithms did not use it and relied completely on the CPU for execution. On the other hand, projects that incorporated DL had powerful combinations of CPUs and GPUs; and the deep learning models were optimized to run in parallel which successfully reduced their processing time. Supervised/unsupervised techniques took the most time because one of the projects (Wu et al., 2020), used unsupervised classifiers to analyze 2D and 3D information. Taking 60 s on average per frame. Without this project, then the average for supervised/unsupervised is of 2.01 s, which would make it the fastest detection methodology category.

5. Produce localization methodologies

Once produce is detected the next step is to compute its spatial location with respect to the robot. This information is needed as it will allow the robot to engage its motion delivery system to move its end-effector towards the produce that needs to be harvested (See Fig. 1). The quality of the information produced by this process will often be reflected in the robot harvesting performance, as there is a direct dependency between the localization accuracy and produce grasping rate (Gongal et al., 2015).

The expected output of the localization process is 3D coordinates that will allow the robot to harvest detected produce by moving its motion delivery system. However, only relying on a 3D position for crops such as sweet pepper, often results in low harvesting success. This is because the end-effector is likely to collide with branches and/or other produce as it reaches its target (Ling et al., 2019). Therefore, researchers conclude that more information (such as produce orientation) from the environment is required to further improve the harvesting rate.

This section describes the most widely applied localization technique in use at harvesting robots (2D to 3D transform). It then discusses the advantages of computing the 3D pose.

5.1. 2D to 3D Transform

If 3D information is available, this technique is generally the simplest method to acquire produce spatial data. It works by using the 2D coordinates produced by the detection phase and correlating them to the depth information provided by the sensor, effectively obtaining the 3D position of the produce (T. Zhang et al., 2020; L.L. Wang et al., 2017; Birrell et al., 2020). A similar process is performed when using a single monocular camera, however as 3D information is not available, it is obtained by knowing the relationship between the focal length of the camera and the pixel size. However, this requires that the distance information is constantly being updated as the robot moves its end-effector. Also, using a monocular camera requires the constant computation of the inverse kinematics for the motion delivery system.

Table 5
Vision subsystem hardware characteristics by detection methodology. After grouping by methodology, the average, precision, recall, F1 score, and time for detection were calculated. Hardware selection was done by CPU characteristics.

Detection Methodology	# Of Projects	Hardware		Average Precision [%]	Average Recall [%]	Average F1 Score [%]	Average Time [s]
		Low-end	High-end				
Single/Multiple Feature	3	Intel i5 3230 M	Intel i5 6700	81.66	60	NA	5.26
Supervised/ Unsupervised	7	Intel i5 2500	Intel Xeon E5 1620	87.64	87.03	85.94	10.09
Deep Learning	8	Intel i5 8400/NVIDIA Tesla P40	Intel Xeon E3 1245/NVIDIA GTX Titan	92.055	90.74	90.26	2.29
Multiple	2	Intel i5 3210 M/NVIDIA GTX 1060	Intel i7 4500U	98.3	88.65	91.18	6.29

Potentially slowing down the overall harvesting speed (Gongal et al., 2015).

Lin et al. (2019b) use this technique to compute the 3D localization of guava by using an active RGB-D camera. First, a colored image is color segmented by a CNN, then a 2D clustering algorithm is used to detect individual fruits and generate bounding boxes. Finally, the bounding boxes 2D center is used to obtain the matching pixel's depth information. Arad et al. (2020) used a CNN to segment pepper from RGB-D images. Then, the 3D location was calculated by using the point of mass of the detected pepper and its direct 2D to 3D transformation to the depth information.

5.2. 3D pose produce estimation

Recently, researchers have found a correlation between robot harvesting performance and the availability of 3D pose information. If only the fruit position is available, the end-effector is likely to collide with branches of the crop as it moves towards the fruit (Ling et al., 2019). Lowering the harvest success rate (see Table 1) (Lin et al., 2019b). Therefore, researchers argue that pose estimation for fruit and peduncle is as important as fruit detection (Lin et al., 2019b). As it will enable the robot to correctly guide its end-effector without collision for optimal cutting.

Six degrees (6D) pose estimation techniques have long been an area of interest in computer vision for object-picking robots (Y. Li et al., 2020). However, the applications of these techniques in harvesting robots are still growing. Most of the robots presented here perform localization by only computing the 3D spatial location without orientation. And the projects that presented pose estimation reported methodologies that did not use DL, such as symmetry plane analysis and shape matching. Techniques that often require a combination of 3D spatial data from the scene and traditional machine learning techniques to identify different parts of a crop.

6. Eye-Hand coordination

Once ripe produce has been detected and 3D localized with respect to the robot, the next step for the robot is to actuate its motion delivery system and move its end-effector towards the produce to harvest. Here, the robot's hardware relies on the information of its vision system to perform movements and guide its motion. This is called eye-hand coordination. Most harvesting robots use an open-loop eye-hand coordination control system. Open-loop depends directly on the accuracy of the vision system that is utilized. An alternative to this framework is the use of a visual feedback control loop also called visual servoing. Which constantly uses visual features to dynamically control the pose of its end-effector (Zhao et al., 2016). This section presents an overview of the eye-hand coordination control at harvesting robots.

6.1. Open-loop visual control

Open-loop visual control employs the "looking then moving" mode of operation. This operation mode relies on the quality of 3D information available from the vision system. By using the 3D information to obtain produce distances, it then calculates the trajectory of the manipulator by calculating the kinematics of the robot. Because of this, the kinematic model and the vision system have to be very accurate (Zhao et al., 2016).

This control scheme is the most widely used across state-of-the-art harvesting robots. Onishi et al. (2019) first performed apple recognition. Once it had the 3D position available it calculated the inverse-kinematics once, so that the robotic arm was able to harvest. Once (T. Zhang et al., 2020) successfully identified the peduncle, they used a CNN to calculate the inverse kinematics of their robot. The output of this CNN was then used to successfully guide the robotic arm to the produce and then harvest it.

A benefit of this system is that it is generally faster than visual servoing (next section) as it only needs to calculate the trajectory once. However, the framework does not respond well to perturbances of the canopy (e.g., canopy movement by the manipulator or by the wind). When these situations occur, the efficiency of this framework for harvesting robots is very low (Zhao et al., 2016). Another limitation is that the motion delivery system must be very precise, which often elevates the hardware costs. If the precision of the delivery system is not enough as well as the resolution and calibration of the vision system, the harvesting success of the robot will be affected.

6.2. Visual servo control

In contrast to open-loop, the input to visual servo control is continuous and dynamic. Its main objective is to estimate the pose of the end-effector using visual features from the sensor and vision controllers. To perform this process, the vision sensor is often attached to the motion delivery subsystem. Its main advantages are that it does not require a precise kinematic model of the robot nor calibration of the vision system (Zhao et al., 2016). Another benefit is that the requirement of having a very precise motion delivery system can be eliminated. However, a major limitation is that it is often slower, as it requires constant computation for the end-effector pose estimation. The average time for robots that used this technique at Table 1 was 19.5 s, which is higher than the average for projects that open-loop visual control (12.26 s).

Arad et al. (2020) used this approach to successfully guide their end-effector towards sweet peppers that needed harvesting. Their robot first identified the sweet peppers by color, then, the motion delivery system was used in a closed-loop with the vision system information to reach the peduncle. On average, the robot took 24 s per pepper. Barth et al. (2016) used an eye in hand (camera at the end-effector) and visual servo framework to successfully approach produce with dense vegetation.

If the 3D location can be accurately estimated at the beginning of the manipulator movement, the harvesting robot can avoid the use of visual servoing while potentially improving produce reaching speed (Gongal et al., 2015)

7. Current and future trends

After analyzing multiple research projects, it was determined that the recent advances in technology and the adoption of Artificial Intelligence detection techniques have increased the produce detection performance of harvesting robots. Unfortunately, further research and development are still needed before a harvesting robot achieves the same performance as current human harvesters: while state-of-the-art harvesting robots vision systems produce detection has increased thanks to the adoption of artificial intelligence techniques, current vision systems still struggle to accurately localize produce and, because of this, harvesting robots still offer low-performance indicators in unmodified production environments. For the harvesting cycle and harvesting success to increase, the average detection and localization time need to decrease while still offering high accuracy. Especially for localization, as it was found that even though vision systems can successfully detect produce, most harvesting errors were due to either inaccurate localization information or obstacles when trying to perform the harvesting actions. Because of these, more research is needed in localization techniques, such as pose estimation or produce searching algorithms through highly occluded environments without compromising on testing new object detection algorithms, as well as new hardware such as new sensors or hardware-accelerated single board computers for produce detection.

This section discusses current and future trends that were identified by analyzing the current state of harvesting robot's vision subsystems as well as the author's recommendations and perspectives on how to solve the current problems that directly impact the vision subsystem and the harvesting robot's performance metrics.

7.1. Increasing produce detection and localization performance

As mentioned, there have been multiple advances made in recent years by using Artificial Intelligence methodologies, unfortunately, the harvesting robots analyzed here still present low harvesting success when performing in unmodified production environments. The highest harvesting success found in a robot was 97.1 percent (see Table 1), however, the authors stated that the harvesting success was in a range of 5% to 97.1%. Where the low performance was attributed to the false positives that occurred when detecting ripe fruits: the result of unpredictable lighting conditions and/or highly occluded environments (clusters of fruit, branches, or leaves) (Gongal et al., 2015). This problem is heightened, when produce requires to be harvested by cutting the peduncle, as the robot needs to estimate the pose of the peduncle which is often small and occluded by other parts of the canopy. These variables result in errors even after the fruit is correctly identified and located: the end effector collides with other fruits, branches, or foliage from the same tree (Kapach et al., 2012), or the cutting point is either not correctly detected or the robot spends a large amount of time searching for it (T. Zhang et al., 2020; Arad et al., 2020). While the overall detection performance has improved when compared with previous works (Bac et al., 2014), there is still room for improvement in both, detection and especially localization by implementing or improving produce recognition vision systems algorithms.

As such, researchers have used many different algorithms to process the sensor information, with a clear trend of relying more on machine learning-based techniques such as deep learning, which was the most popular technique used in the analyzed works (25% of projects). The interest in applying deep learning algorithms for produce detection is that works have shown it has increased the detection performance even on very difficult scenes such as the ones with variable lighting, multiple occlusions, and even clusters of the same produce. Unfortunately, when compared with other techniques such as unsupervised classifiers or single/multiple feature analysis, these techniques require big, labeled datasets, powerful computers with dedicated GPUs to achieve real-time performance, and long periods to train the deep learning detection model (sometimes even weeks). As such, even though deep learning techniques have shown great promise for produce detection, as discussed, there are still works that utilize other algorithms to detect produce.

Still, the use of deep learning techniques is growing for produce detection and localization. And as such, more research is still needed. This section discusses trends that directly impact produce detection and localization metrics, as well as our recommendations and perspective on how to solve the current recognition problems.

7.1.1. Data augmentation

Current produce detection methodologies that involve deep learning techniques require large manually labeled datasets that closely match the production environment conditions where the vision subsystem is intended to run. Fortunately, today there are many public datasets that researchers have published to test the performance of new detection algorithms for popular fruits and vegetables. Unfortunately, there are many different fruits and vegetables across the world, and creating/maintaining many large datasets of high-resolution data with many different scene conditions would be very difficult. Not to mention the amount of manual and repetitive effort involved when creating these datasets. On top of that, the dataset labeling process is also prone to human error, as some objects might not be correctly labeled, some objects might be completely missed, and even the characteristics of the labeled objects might differ from one human labeler to another one.

To increase the performance of models trained with small datasets, researchers have augmented the data by using proven and common transformations such as spatial flipping, image warping, brightness manipulation, etc. Some have even used more complex techniques such as the one used by Su et al. (2021) where they create one new image by combining smaller sections of six other images. Reporting detection performance increases by at least 3%. However, the datasets on which this technique was tested still had more than 140 images each. Which still required significant effort and time to create. A possible solution that would facilitate the creation of datasets for produce detection is the use of few-shot or few-example object detection frameworks such as the ones implemented and tested by Dong et al. (2019), Fu et al. (2019) and Xiao et al. (2021). Where the authors successfully detected novel objects with a small set of labeled images (4–10). Albeit, with less accuracy than larger datasets. On the other hand, the recent availability of photo-realistic simulators such as NVIDIA Isaac Sim (NIVIDA, 2021) will facilitate researchers in the creation of realistic synthetic datasets which can accurately simulate illumination conditions at different times of the day. This has the added benefit that detection and localization algorithms can also be tested under many different user-set conditions.

The use of multisensory systems can also help to increase the detection and localization performance without the need for complex deep learning methodologies. As shown in the work of (Fernández et al., 2014), where the fusion of multiple features extracted from color, multispectral, and ToF cameras greatly increased the differences between information gathered from parts of a crop and its produce. With the main benefit of no longer requiring any pre-treatment on the images before processing. Unfortunately, vision systems such as this one, are very tailored to the reflectance of specific produce and might not achieve the same performance on other fruits and vegetables; even after adjusting the detection parameters.

7.1.2. Deep learning model optimization

As discussed, DL detection methodologies perform better than traditional detection techniques; even on challenging scenarios such as ones with multiple shadows, diverse lighting conditions, and multiple occlusions. Unfortunately, this performance improvement also requires the use of high-performance computers with dedicated GPUs. Which are necessary to improve the inference time when detecting objects, as DL models are very computationally expensive. One characteristic of DL models is that they tend to use many hidden layers, which can make the model very complex, and thus, require powerful hardware for training and inference.

While current experiments have shown that detection performance has increased, the average detection time per frame of the projects analyzed here is still very large: 6.11 s. Even with some vision subsystems relying on the use of powerful computer hardware. One possible solution to this issue is the use of optimized models such as CSL-YOLO, proposed by Zhang et al. (2021). Which generates redundant features used for object detection with cheap operations. Effectively reducing the amount of computational power required by 43%. Unfortunately, YOLO-based architectures tend to struggle when detecting small objects, and as such would only be useful when detecting big produce (such as apples, oranges, eggplants, etc.) or by processing close-up photos of smaller produce (such as berries) (Mazzia et al., 2020).

Another possible solution is the optimization of object detection models with the use of optimization frameworks such as TensorRT by NVIDIA. Which can accelerate the time performance by combining the layers of a model, effectively improving throughput, power efficiency, and memory consumption (Kalampokas et al., 2021). Badeka et al. (2021) optimized EfficientDet-D0, an artificial neural network model using TensorRT achieving a mean average precision (mAP) of 77.9% for detecting vineyard trunks in 38 ms. However, one downside of this method is that TensorRT optimizations are tailored to the hardware where the model is intended to run, and as such, can create compatibility issues once the hardware is no longer available.

7.1.3. Pose estimation

Regarding produce localization, researchers have concluded that having pose information is important for harvesting robots. Especially for produce that must be harvested by cutting the peduncle (such as

sweet pepper), as the end-effector and motion delivery subsystems must account for the inclination of the produce. Today, pose estimation is a research hotspot for retail robot pickers in the field of computer vision (Zhuang et al., 2021; Guo et al., 2019). Unfortunately, those techniques are not directly compatible with fresh produce, as those techniques require known sizes for correct pose estimation. In addition, these techniques were also tested in idealized conditions (controlled lighting and unblocked views of the products. And while there are production environments where lighting can be controlled, crops still grow various shapes and sizes, which can cause shadows and obstructions by parts of the plant.

As such, current research has mostly focused on using 3D information to match 3D points to known models of the produce. Allowing the vision subsystem to estimate 3D pose correctly. This method was used by Sa et al. (2017) showing great accuracy for detecting sweet pepper peduncles and their poses. However, the main downside of this method is that it requires an accurate preferably close-up 3D visualization of the peppers, as the algorithm processing time increases drastically when the number of 3D points available increases. As will be discussed in the next section, today there are a variety of sensors that are able to provide 3D information; being RGB-D cameras the current most popular choice. Unfortunately, in the authors' experience, these sensors require a minimum distance to provide accurate depth results. And when the sensor has enough distance to the crop's canopy, there is a lot of noise in the depth information because of the cluttered canopy of sweet pepper. This noise combined with the low resolution of current sensors does not allow for accurate 3D model matching unless the sensor is at a close distance as shown by Sa et al. (2017).

A possible solution is to perform simultaneous localization and mapping (SLAM) for high-density vegetation crops as shown by Barth et al. (2016). This allows the robot to recreate an accurate 3D representation of the crop that is being harvested by performing a close-up scan of the plant. Which could even allow for the implementation of searching algorithms for produce that is heavily occluded in the canopy. Much like human harvesters tend to look for occluded produce.

Unfortunately, crop parts tend to move easily when disturbed which introduces new potential problems, such as visual tracking when produce sways. This problem becomes even more critical when the production environment is outdoors such as a harvesting robot working at an orchard as shown by Q. Yang et al. (2020). As of now, the robot's end-effector subsystem needs to match the swaying of the produce. Introducing then the necessity of tracking algorithms in the vision subsystem to perform visual servoing.

7.2. Technological adoption

As new lower-cost sensors and computer hardware are available, researchers will be able to test more robust produce detection techniques. Regarding sensors, a tendency towards using an RGB-D-based sensor in current produce detection vision subsystems was observed, specifically active camera-based sensors. This can be explained by the benefits these sensors provide: fused color and depth information and a good price-performance ratio compared to other 3D-capable sensors such as LiDARs. This allows the use of 2D images and 3D data processing techniques which have shown better performance when operating under highly occluded scenarios or with same-colored overlapping produce. The use of RGB-D sensors can also be attributed to the need to obtain better quality information to compute produce localization without the need for an additional sensor. Having 3D information can also facilitate the process of produce pose estimation which can help increase the robot's harvesting success. However, there is still room for improvement regarding RGB-D sensors in the hardware and the algorithms. As active sensors are particularly sensitive whenever exposed to direct sunlight and the depth resolution is relatively low when compared to the 2D image they can provide (Fu et al., 2020). In addition, in our experience, active RGB-D-based sensors, tend to require a minimum distance from the sensor to the object in order to report accurate depth information. Unfortunately, if the sensor's distance is too big, the depth information tends to have a lot of noise, as most produce production environments tend to have multiple objects in a scene such as branches, leaves, and even other produce. This is also heightened by the different reflective surfaces and orientations that can be found in a crop's canopy. As such, we suggest that when working with these sensors, a two-step approach is taken for produce detection and localization: (1) first the robot should try to detect available produce by only relying on the colored image. (2) Then, using the coordinates of the inferred produce, the robot should try approaching the produce to a distance where the sensor provides good quality depth information. This would allow the vision subsystem to acquire good colored and depth data from the produce, which could potentially increase the harvesting robot's performance indicators.

As mentioned, there is a noticeable trend towards using deep learning methodologies in harvesting robots. However, this technique still requires large manually annotated datasets and powerful computers with dedicated GPUs for speeding up the training and inference processes. Even though training time is not as critical as the inference time for the harvesting robot, it is still important to reduce it. As it might speed the testing process with different crops. Another limitation of this technique is that the speed of training and inference often depends on the capabilities of the computer that is being used. If inference speed is to be reduced, a more powerful computer is required. Which can greatly increase the costs of the vision subsystem alone. While the introduction of Edge AI embedded hardware accelerators such as the NVIDIA Jetson family can provide the same level of accuracy while using less power, a powerful computer is still required for training. Not to mention the amount of time to create the dataset by manually annotating each image. However, thanks to the availability of cloud GPUs, there might be a possible budget-friendly workflow that uses the cloud GPUs for training the model, and then an edge AI accelerator for deployment. This workflow has the potential of being cheaper than building a powerful computer for model training and then using the same system on the robot for inference.

While the adoption of Edge AI in agriculture is still occurring (Only 1 out of the 30 projects is currently using an edge AI accelerator), it is showing promising results. As mentioned, one of the main benefits of edge AI, when compared against high-speed wireless connections, is that inference will not be interrupted by wireless interference. As it is performed onboard.

On the other hand, the deployment of reliable, high-throughput wireless connections such as 5G might allow for more cost reduction in the vision subsystem. The introduction of 5G will allow to process images or video feeds on the cloud. Successfully offloading the inference process from the harvesting robot without losing real-time performance. However, the adoption of 5G still has a long way to go, as it is currently still being deployed across the world. Perhaps one solution that can be easily adapted when 5G is successfully deployed, is the deployment of a WiFi6 network on the production environment, and the use of a powerful server that constantly processes data sent by multiple robots. In a similar fashion as today's logistic fleets of robots in warehouses.

Regarding the motion-delivery subsystem, most harvesting robots are using off-the-shelf anthropomorphic robotic arms. This allows for a faster and more reliable hardware architecture, as these systems are already tested and are very precise. Unfortunately, the harvesting success is still very low due to poor quality localization information, or because the end-effector collides/gets stuck with parts of the canopy. One possible research avenue is the use of visual servo control and the use of 3D SLAM for produce detection, as it can allow for the use of a less precise motion delivery subsystems, which could potentially lower the costs of the harvesting robot without sacrificing the harvesting performance.

7.3. Multiple subsystems collaboration

Harvesting robots are a complex problem that requires multiple subsystems working together.

The vision subsystem is of critical importance as it will enable the robot to visualize its working environment and thus, detect and localize ripe produce. Once produce is detected and localized, the vision subsystem will provide the information that the motion delivery subsystem needs to guide the end-effector towards the produce for harvesting. While the focus of this work was to analyze the performance of the current harvesting robot's vision subsystems, the overall harvesting success of the robot depends on having accurate produce information, precise, agile, and reliable motion delivery, and end-effector subsystems that can accommodate multiple sizes and shapes of produce. As such, it is increasingly difficult to measure the harvesting performance of a robot by only taking into consideration each subsystem separately. Today a harvesting robot can have multiple sensors in any subsystem. Currently, harvesting robots might have different vision sensors onboard, some on the main body/chassis, and others on the end-effector as well as on the robotic arm. This section presents some subsystems collaborations that if implemented can help harvesting robots to improve their harvesting performance metrics.

Today, perhaps the most widely subsystem collaboration research is the use of visual servo control. Where a vision sensor is mounted at the end of the motion delivery subsystem and is able to provide closed-loop feedback regarding the position of the end-effector and the detected produce. This technique is currently being used by not more than 6.5% of the projects analyzed here. However, most robots still rely on open-loop visual control. Both have their advantages and limitations, however, where visual servo control excels, is that there is constant communication between the state of the motion delivery subsystem and the produce detection subsystem. Which as mentioned, allows for less precise motion delivery subsystems which can lower the overall harvesting robot cost. Another advantage of this system is that a more accurate representation of the scene where produce was identified can be created. This could potentially allow the motion delivery subsystem to evade obstacles such as branches, leaves, or even other produce when performing the harvesting action (much like a human harvester); and if feedback sensors such as resistive touch sensors in some areas of the arm, were added to the motion delivery subsystem, it could sense if it got stuck. Much like a human harvester would search for produce that is occluded in the canopy. Effectively reducing the damage to the crop's canopy and enabling the robot to perform produce search in high-density canopies. Unfortunately, no project reported the ability to perform this process.

A possible explanation on why this ability has not been developed is that current research projects are still focusing on improving the detection and localization performance of produce that is clearly visible. In other words, produce that can be harvested without the need of searching algorithms in dense vegetation. On the other hand, maneuvering through a dense canopy presents multiple opportunities for the motion delivery to get stuck and even damage the crop's canopy if it doesn't have adequate sensors or information about the canopy. Both of which would increase the cost of the final harvesting system. Because of this, researchers have concluded that it might be easier to train or breed plants in such a way that more fruit is grown at the front of the plant (Bac et al., 2016), meaning produce should become more accessible to the harvesting robots.

Other research areas involve the collaboration of the vision subsystem and motion delivery subsystem to search for produce that is not clearly visible from outside the canopy. Possible implementations might even require a robot with two collaborative arms, one for removing the obstacles and the other to harvest produce. As well as implementing a vision subsystem capable of identifying as obstacles leaves, branches, trunks, and even other produce that is not ready to harvest. Effectively mimicking the human harvesting process and reducing the robot's harvesting cycle. As multiple processes will be performed in parallel: obstacle removal, produce detection and localization, and motion delivery subsystem movement towards ripe produce.

As mentioned, harvesting robots might provide a solution to the current labor shortage in the agricultural industry. However, harvesting robots as well as their subsystems still need substantial improvement to match the speed of human harvesters. As such, another research area is the use of collaborative robots for harvesting specialty crops: humans and robots working together to harvest produce. Which might be able to lower the impact of the current workforce shortage. As a first step, this collaboration could start with robots that assist with the transport of crops/produce across the production environment. Then this could evolve to robots that are able to assist humans with preventive or maintenance work at the plants: such as trimming the leaves of tomato plants for harvesting.

7.4. Current and future trends diagram

Table 6 presents a simple view of the current and feature trends that were identified and discussed in Section 7. This table describes how the current trend is being performed and how most likely it will be done in the near future.

Table 6
Current and Future Trends Diagram.

Trends	Current	Future
Data Augmentation	Manual process, requires large amounts of data time and is prone to human error.	Data augmentation techniques such as few-shot or few-example.
Deep Learning Model Optimization	Large and complex deep learning models, that require powerful hardware for inference.	Small optimized deep learning models or model optimization through libraries.
Eye-Hand Coordination	Open-loop visual control is the most popular approach. Requiring very precise vision sensor and motion delivery subsystems.	A transition towards visual servo control. allowing for produce search algorithms through the canopy and the use of less precise motion delivery subsystems.
Pose Estimation	Pose estimation of peduncle via fusion of 2D and 3D information.	Pose estimation and SLAM through 2D and 3D information.
Sensor Adoption	Detection and pose estimation are done with different sensors.	Use an RGB-D-based sensor for detection and localization.
Cloud Training Infrastructure	Training is done on-site. Requires acquisition of powerful and expensive computers.	Use pay-as-you-go cloud training infrastructure.
Edge AI Adoption	Robots have powerful full-size computers onboard.	Adoption of Edge AI hardware accelerator embedded devices.
Obstacle Evasion	Use of 3D information for a simple obstacle avoidance system.	Use of 3D information and feedback sensors on robotic arms to sense obstacles.
Ripe Produce Search	Robots identify occluded produce, but do not attempt to harvest it.	Multiple robots' collaboration for canopy obstacle removal and ripe produce search.
Collaborative Robots	A complete harvesting system works in a production environment without any human intervention.	Multiple robot collaboration and/or human-robot collaboration.

8. Conclusions

In this survey, a review of the current methodologies for produce detection and localization was provided by analyzing different harvesting robots and vision subsystems research projects published in the last few years. In addition, in this study, an in-depth analysis of the hardware and software that current vision subsystems for produce detection are using was also presented. The work resulted in the following findings for produce detection: the average precision was 90.26%, the recall was 88.06 %, the F1 score of 88.59 %, and the average detection time was of 6.13 s. For produce localization: the success rate was 76.62%, 4 works performed pose estimation, and the average time for localization was of 4.8 s. However, localization metrics were too sparse as more than 60% of the works are focusing only on produce detection, and as such did not perform produce localization; and from the ones that did, unfortunately, most did not report localization metrics. For the motion delivery subsystem: the most popular architecture was the anthropomorphic arm with >5 DoF and only 6% of the works performed visual servoing. Suggesting the importance of maneuverability as part of the harvesting process. Finally, for the end-effector subsystem, the most common harvesting method was found to be grasping with 20% followed by grasping by 6%. Regarding main sensor for the vision subsystem: 16% used binocular RGB-D cameras, 30% active RGB-D cameras, 3% LiDARs, 3% multi-spectral cameras, 36% RGB cameras, and 10% multisensory systems.

Unfortunately, most projects did not report the complete metrics that were proposed in this manuscript. Nevertheless, the information presented here is a general representation of the current state of the harvesting robot's vision subsystems. Suggesting that the recent technological advancements and the use of artificial intelligence techniques have allowed vision systems to improve with respect to the findings of (Bac et al., 2014) in nearly all indicators except for produce localization success rate: 76.62% against 85%. This could be explained by the recent adoption of new sensors for localization such as RGB-D and that only a small amount of the projects analyzed here were tested in structured environments such as laboratories and greenhouses. Which, as expected reduced the localization performance of projects tested outdoors due to factors such as the movement caused by wind, and the reduced accuracy of active RGB-D sensors when used under direct sunlight.

The increase in the detection indicators can be attributed to the adoption of more artificial intelligence techniques such as deep learning. Of the research projects analyzed, deep learning methods were the most popular produce detection methodologies with more than 25%. As mentioned, the main benefit of these techniques is that they are more robust when analyzing complex scenes such as varying lighting, which can drastically change the color of the produce at the images and can also cause multiple shadows; and highly occluded scenes, where the produce is not clearly visible and is obstructed by either more produce or parts of the crop canopy. The reason for this performance increase is because deep learning models do not use manual feature extraction. DL uses layers of neural networks that essentially extract features (patterns, texture, shapes, color distribution, etc.) automatically. Whereas traditional techniques only rely on analyzing one manually extracted feature, which due to the morphological characteristics of the crop, has multiple variations even on the same produce. In addition, deep learning techniques show great promise when analyzing different images where produce appears at different locations as well as in different sizes. Unfortunately, the use of these techniques requires powerful hardware for fast training and inference which often is expensive and not available. And unfortunately, the speed will be greatly affected if deployment of the trained model is not done on a computer with similar hardware specifications to the one in which it was trained. In other words, these methodologies require additional rework when the hardware is unavailable.

Even though the field of produce detection is currently applying multiple artificial intelligence methodologies, research in other techniques such as color and shape detection (single/multiple feature analyses) is still being used: these methodologies were used by 10% of the analyzed projects. Luo et al. (2018) reported a grape detection accuracy of 81.66% by only using a simple color threshold algorithm. This has the benefit of achieving relatively good performance with fast and simple algorithms that do not require a labeled dataset and powerful and costly hardware such as GPUs.

Despite the fact that the use of multiple detection methodologies as the main produce detection algorithm was the least used methodology (6% of researched projects), their indicators outperformed those of deep learning methodologies (See Table 5). The reason for this is that they combined the two most widely adopted methodologies: deep learning and supervised/unsupervised classifiers for result refinement. Which as shown, can increase the detection performance.

While many different combinations of detection/localization methodologies and hardware for harvesting robots' vision subsystems have been published, there is still room for more research as current harvesting robots are not able to achieve a 100% harvesting success rate. While there are multiple complex subsystems in a harvesting robot, the vision subsystem is critical, as it will provide all the necessary information to the motion delivery and end-effector subsystems. There is still room for improvement regarding the vision subsystem's performance indicators, however, the analysis provided here has shown the potential that artificial intelligence methods and technological advancements can provide to the agriculture industry. Sensors such as multispectral cameras allow for an objective quality assessment of fruits and vegetables by using wavelengths that can penetrate the produce (Lu et al., 2017). A process that cannot be done by human harvesters. On the other hand, harvesting robots have the advantage that they can work continuously without breaks provided there are enough numbers of them and/or a constant power supply, and they also have the potential of being very efficient and precise by only harvesting ripe produce that meets quality standards that were previously set. However, current harvesting robots for precision agriculture have yet to match the speed of human harvesters, especially for produce that must be cut from the peduncle and is often occluded.

In this study, an overview of how recent artificial intelligence methodologies and new hardware has impacted current vision subsystems' performance of harvesting robots was presented. As a result, an analysis of the most popular vision subsystems of harvesting robots as well as the most popular vision sensors and detection and localization methodologies was provided. Finally, current and future trends were identified and provided. Discussing current challenges that still require further research for the robots' harvesting performance to improve. Essentially providing some areas for future research.

CRediT authorship contribution statement

Luis-Enrique Montoya-Cavero: Conceptualization, Investigation, Writing – original draft. **Rocío Díaz de León Torres:** Conceptualization, Writing – review & editing. **Alfonso Gómez-Espinosa:** Writing – review & editing. **Jesús Arturo Escobedo Cabello:** Conceptualization, Project administration, Supervision, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to acknowledge CONACyT for the support in the postgraduate studies of the first author (scholarship CVU #902093).

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Submission declaration and verification

The work described has not been published previously and it is not under consideration for publication elsewhere. This work publication is approved by all authors and, if accepted, it will not be published elsewhere in the same form.

References

Arad, B., Balendonck, J., Barth, R., Ben-Shahar, O., Edan, Y., Hellström, T., Hemming, J., Kurtser, P., Ringdahl, O., Tielen, T., van Tuijl, B., 2020. Development of a sweet pepper harvesting robot. Journal of Field Robotics 1–13. https://doi.org/10.1002/rob.21937. November 2019.

Arad, B., Kurtser, P., Barnea, E., Harel, B., Edan, Y., Ben-Shahar, O., 2019. Controlled lighting and illumination-independent target detection for real-time cost-efficient applications. The case study of sweet pepper robotic harvesting. Sensors (Switzerland) 19 (6), 1–15. https://doi.org/10.3390/s19061390.

Astill, G., Perez, A., Thornsbury, S., 2020. Developing Automation and Mechanization for Specialty Crops : A Review of U.S. Department of Agriculture Programs A Report to Congress.

Bac, C.W., Roorda, T., Reshef, R., Berman, S., Hemming, J., van Henten, E.J., 2016. Analysis of a motion planning problem for sweet-pepper harvesting in a dense obstacle environment. Biosyst. Eng. 146, 85–97. https://doi.org/10.1016/j.biosystemseng.2015.07.004.

Bac, C.W., van Henten, E.J., Hemming, J., Edan, Y., 2014. Harvesting robots for high-value crops: state-of-the-art review and challenges ahead. J. Field Rob. 31 (6), 888–911. https://doi.org/10.1002/rob.21525.

Badeka, E., Kalampokas, T., Vrochidou, E., Tziridis, K., Papakostas, G.A., Pachidis, T.P., Kaburlasos, V.G., 2021. Vision-based vineyard trunk detection and its integration into a grapes harvesting robot. Int. J. Mech. Eng. Rob. Res. 10 (7), 374–385. https://doi.org/10.18178/ijmerr.10.7.374-385.

Baeten, J., Donné, K., Boedrij, S., Beckers, W., Claesen, E., 2008. Autonomous fruit picking machine: a robotic apple harvester. Springer Tracts Adv. Rob. 42, 531–539. https://doi.org/10.1007/978-3-540-75404-6_51.

Barth, R., Hemming, J., van Henten, E.J., 2016. Design of an eye-in-hand sensing and servo control framework for harvesting robotics in dense vegetation. Biosyst. Eng. 146, 71–84. https://doi.org/10.1016/j.biosystemseng.2015.12.001.

Birrell, S., Hughes, J., Cai, J.Y., Iida, F., 2020. A field-tested robotic harvesting system for iceberg lettuce. J. Field Rob. 37 (2), 225–245. https://doi.org/10.1002/rob.v37.210.1002/rob.21888.

Bonadies, S., Gadsden, S.A., 2019. An overview of autonomous crop row navigation strategies for unmanned ground vehicles. Eng. Agric. Environ. Food 12 (1), 24–31. https://doi.org/10.1016/j.eaef.2018.09.001.

Bulanon, D.M., Burks, T.F., Alchanatis, V., 2009. Image fusion of visible and thermal images for fruit detection. Biosyst. Eng. 103 (1), 12–22. https://doi.org/10.1016/j.biosystemseng.2009.02.009.

Ceres, R., Pons, J.L., Jiménez, A.R., Martín, J.M., Calderón, L., 1998. Design and implementation of an aided fruit-harvesting robot (Agribot). Ind. Rob.: Int. J. 25 (5), 337–346. https://doi.org/10.1108/01439919810232440.

Chaivivatrakul, S., Dailey, M.N., 2014. Texture-based fruit detection. Precis. Agric. 15 (6), 662–683. https://doi.org/10.1007/s11119-014-9361-x.

Chakraborty, M., Khot, L.R., Sankaran, S., Jacoby, P.W., 2019. Evaluation of mobile 3D light detection and ranging based canopy mapping system for tree fruit crops. Comput. Electron. Agric. 158 (October 2018), 284–293. https://doi.org/10.1016/j.compag.2019.02.012.

Chen, C., Li, B., Liu, J., Bao, T., Ren, N., 2020. Monocular positioning of sweet peppers: an instance segmentation approach for harvest robots. Biosyst. Eng. 196, 15–28. https://doi.org/10.1016/j.biosystemseng.2020.05.005.

Dong, X., Zheng, L., Ma, F., Yang, Y.i., Meng, D., 2019. Few-example object detection with model communication. IEEE Trans. Pattern Anal. Mach. Intell. 41 (7), 1641–1654. https://doi.org/10.1109/TPAMI.3410.1109/TPAMI.2018.2844853.

Drury, B., Valverde-Rebaza, J., Moura, M.F., de Andrade Lopes, A., 2017. A survey of the applications of Bayesian networks in agriculture. Eng. Appl. Artif. Intell. 65 (July), 29–42. https://doi.org/10.1016/j.engappai.2017.07.003.

Edan, Y., Miles, G.E., 1994. Systems engineering of agricultural robot design. IEEE Trans. Syst. Man Cybern. 24 (8), 1259–1265. https://doi.org/10.1109/21.299707.

Eizentals, P., Oka, K., 2016. 3D pose estimation of green pepper fruit for automated harvesting. Comput. Electron. Agric. 128, 127–140. https://doi.org/10.1016/j.compag.2016.08.024.

Feng, J., Zeng, L., He, L., 2019. Apple fruit recognition algorithm based on multi-spectral dynamic image analysis. Sensors (Switzerland) 19 (4), 1–13. https://doi.org/10.3390/s19040949.

Feng, Q., Zou, W., Fan, P., Zhang, C., Wang, X., 2018. Design and test of robotic harvesting system for cherry tomato. Int. J. Agric. Biol. Eng. 11 (1), 96–100. https://doi.org/10.25165/j.ijabe.20181101.2853.

Fernandez, R., Montes, H., Surdilovic, J., Surdilovic, D., Gonzalez-De-Santos, P., Armada, M., 2018. Automatic detection of field-grown cucumbers for robotic harvesting. IEEE Access 6, 35512–35527. https://doi.org/10.1109/Access.628763910.1109/ACCESS.2018.2851376.

Fernández, R., Salinas, C., Montes, H., Sarria, J., 2014. Multisensory System for Fruit Harvesting Robots. Experimental Testing in Natural Scenarios and with Different Kinds of Crops. July 2015. https://doi.org/10.3390/s141223885.

Fountas, S., Mylonas, N., Malounas, I., Rodias, E., Santos, C.H., Pekkeriet, E., 2020. Agricultural robotics for field operations. Sensors (Switzerland) 20 (9), 1–27. https://doi.org/10.3390/s20092672.

Fu, K., Zhang, T., Zhang, Y., Yan, M., Chang, Z., Zhang, Z., Sun, X., 2019. Meta-SSD: towards fast adaptation for few-shot object detection with meta-learning. IEEE Access 7, 77597–77606. https://doi.org/10.1109/Access.628763910.1109/ACCESS.2019.2922438.

Fu, L., Gao, F., Wu, J., Li, R., Karkee, M., Zhang, Q., 2020. Application of consumer RGB-D cameras for fruit detection and localization in field: a critical review. Comput. Electron. Agric. 177 (July), 105687. https://doi.org/10.1016/j.compag.2020.105687.

Gan, H., Lee, W.S., Alchanatis, V., Ehsani, R., Schueller, J.K., 2018. Immature green citrus fruit detection using color and thermal images. Comput. Electron. Agric. 152 (March), 117–125. https://doi.org/10.1016/j.compag.2018.07.011.

Gan, H., Lee, W.S., Alchanatis, V., Abd-Elrahman, A., 2020. Active thermal imaging for immature citrus fruit detection. Biosyst. Eng. 198, 291–303. https://doi.org/10.1016/j.biosystemseng.2020.08.015.

Garcia-Lamont, F., Cervantes, J., López, A., Rodriguez, L., 2018. Segmentation of images by color features: a survey. Neurocomputing 292, 1–27. https://doi.org/10.1016/j.neucom.2018.01.091.

Ge, Y., Xiong, Y., From, P.J., 2020. Symmetry-based 3D shape completion for fruit localisation for harvesting robots. Biosyst. Eng. 197, 188–202. https://doi.org/10.1016/j.biosystemseng.2020.07.003.

Gené-Mola, J., Gregorio, E., Guevara, J., Auat, F., Sanz-Cortiella, R., Escolà, A., Llorens, J., Morros, J.R., Ruiz-Hidalgo, J., Vilaplana, V., Rosell-Polo, J.R., 2019. Fruit detection in an apple orchard using a mobile terrestrial laser scanner. Biosyst. Eng. 187, 171–184. https://doi.org/10.1016/j.biosystemseng.2019.08.017.

Gené-Mola, J., Sanz-Cortiella, R., Rosell-Polo, J.R., Morros, J.R., Ruiz-Hidalgo, J., Vilaplana, V., Gregorio, E., 2020. Fruit detection and 3D location using instance segmentation neural networks and structure-from-motion photogrammetry. Comput. Electron. Agric. 169 (January), 105165. https://doi.org/10.1016/j.compag.2019.105165.

Gongal, A., Amatya, S., Karkee, M., Zhang, Q., Lewis, K., 2015. Sensors and systems for fruit detection and localization: a review. Comput. Electron. Agric. 116, 8–19. https://doi.org/10.1016/j.compag.2015.05.021.

GSMA, 2020. Smart Farming: Weed Elimination with 5G Autonomous Robots. February. https://www.gsma.com/iot/wp-content/uploads/2020/02/Smart-Farming-weed-elimination-final-for-web-170220.pdf.

Guo, N., Zhang, B., Zhou, J., Zhan, K., Lai, S., 2020. Pose estimation and adaptable grasp configuration with point cloud registration and geometry understanding for fruit grasp planning. Comput. Electron. Agric. 179, 105818. https://doi.org/10.1016/j.compag.2020.105818.

Guo, Q., Quan, Y., Jiang, C., 2019. Object pose estimation in accommodation space using an improved fruit fly optimization algorithm. J. Intell. Rob. Syst.: Theory Appl. 95 (2), 405–417. https://doi.org/10.1007/s10846-018-0940-3.

Hu, C., Liu, X., Pan, Z., Li, P., 2019. Automatic detection of single ripe tomato on plant combining faster R-CNN and intuitionistic fuzzy set. IEEE Access 7, 154683–154696. https://doi.org/10.1109/Access.628763910.1109/ACCESS.2019.2949343.

Hua, Y., Zhang, N., Yuan, X., Quan, L., Yang, J., Nagasaka, K., Zhou, X.-G., 2019. Recent advances in intelligent automated fruit harvesting robots. Open Agric. J. 13 (1), 101–106. https://doi.org/10.2174/1874331501913010101.

Kadir, M.F.A., Yusri, N.A.N., Rizon, M., bin Mamat, A.R., Makhtar, M., Jamal, A.A., 2015. Automatic mango detection using texture analysis and randomised hough transform. Appl. Math. Sci. 9, 6427–6436.

Kalampokas, T., Vrochidou, E., Papakostas, G.A., Pachidis, T., Kaburlasos, V.G., 2021. Grape stem detection using regression convolutional neural networks. Comput. Electron. Agric. 186, 106220. https://doi.org/10.1016/j.compag.2021.106220.

Kamilaris, A., Prenafeta-Boldú, F.X., 2018. Deep learning in agriculture: a survey. Comput. Electron. Agric. 147 (February), 70–90. https://doi.org/10.1016/j.compag.2018.02.016.

Kang, H., Zhou, H., Wang, X., Chen, C., 2020. Real-time fruit recognition and grasping estimation for robotic apple harvesting. Sensors (Switzerland) 20 (19), 1–15. https://doi.org/10.3390/s20195670.

Kapach, K., Barnea, E., Mairon, R., Edan, Y., Ben-Shahar, O., 2012. Computer vision for fruit harvesting robots – state of the art and challenges ahead. Int. J. Comput. Vis. Robot. 3 (1–2), 4–34. https://doi.org/10.1504/IJCVR.2012.046419.

Koirala, A., Walsh, K.B., Wang, Z., McCarthy, C., 2019. Deep learning – method overview and review of use for fruit detection and yield estimation. Comput. Electron. Agric. 162 (March), 219–234. https://doi.org/10.1016/j.compag.2019.04.017.

Le, T.D., Ponnambalam, V.R., Gjevestad, J.G.O., From, P.J., 2020. A low-cost and efficient autonomous row-following robot for food production in polytunnels. J. Field Rob. 37 (2), 309–321. https://doi.org/10.1002/rob.v37.210.1002/rob.21878.

Lee, W.S., Alchanatis, V., Yang, C., Hirafuji, M., Moshou, D., Li, C., 2010. Sensing technologies for precision specialty crop production. Comput. Electron. Agric. 74 (1), 2–33. https://doi.org/10.1016/j.compag.2010.08.005.

Li, H., Zhu, Q., Huang, M., Guo, Y., Qin, J., 2018. Pose estimation of sweet pepper through symmetry axis detection. Sensors (Switzerland) 18 (9), 1–14. https://doi.org/10.3390/s18093083.

Li, Y., Wang, G., Ji, X., Xiang, Y., Fox, D., 2020. DeepIM: deep iterative matching for 6D pose estimation. Int. J. Comput. Vision 128 (3), 657–678. https://doi.org/10.1007/s11263-019-01250-9.
Lin, G., Tang, Y., Zou, X., Cheng, J., Xiong, J., 2020a. Fruit detection in natural environment using partial shape matching and probabilistic Hough transform. Precis. Agric. 21 (1), 160–177. https://doi.org/10.1007/s11119-019-09662-w.
Lin, G., Tang, Y., Zou, X., Li, J., Xiong, J., 2019a. In-field citrus detection and localisation based on RGB-D image analysis. Biosyst. Eng. 186, 34–44. https://doi.org/10.1016/j.biosystemseng.2019.06.019.
Lin, G., Tang, Y., Zou, X., Xiong, J., Fang, Y., 2020b. Color-, depth-, and shape-based 3D fruit detection. Precis. Agric. 21 (1), 1–17. https://doi.org/10.1007/s11119-019-09654-w.
Lin, G., Tang, Y., Zou, X., Xiong, J., Li, J., 2019b. Guava detection and pose estimation using a low-cost RGB-D sensor in the field. Sensors (Switzerland) 19 (2), 1–15. https://doi.org/10.3390/s19020428.
Lin, P., Lee, W.S., Chen, Y.M., Peres, N., Fraisse, C., 2020c. A deep-level region-based visual representation architecture for detecting strawberry flowers in an outdoor field. Precis. Agric. 21 (2), 387–402. https://doi.org/10.1007/s11119-019-09673-7.
Ling, X., Zhao, Y., Gong, L., Liu, C., Wang, T., 2019. Dual-arm cooperation and implementing for robotic harvesting tomato using binocular vision. Rob. Auton. Syst. 114, 134–143. https://doi.org/10.1016/j.robot.2019.01.019.
Liu, G., Mao, S., Kim, J.H., 2019. A mature-tomato detection algorithm using machine learning and color analysis. Sensors (Switzerland) 19 (9), 1–19. https://doi.org/10.3390/s19092023.
Liu, G., Nouaze, J.C., Mbouembe, P.L.T., Kim, J.H., 2020. YOLO-tomato: a robust algorithm for tomato detection based on YOLOv3. Sensors (Switzerland) 20 (7), 1–20. https://doi.org/10.3390/s20072145.
Lu, Y., Huang, Y., Lu, R., 2017. Innovative hyperspectral imaging-based techniques for quality evaluation of fruits and vegetables: a review. Appl. Sci. (Switzerland) 7 (2), 189. https://doi.org/10.3390/app7020189.
Luo, L., Tang, Y., Lu, Q., Chen, X., Zhang, P., Zou, X., 2018. A vision methodology for harvesting robot to detect cutting points on peduncles of double overlapping grape clusters in a vineyard. Comput. Ind. 99, 130–139. https://doi.org/10.1016/j.compind.2018.03.017.
Luo, L., Tang, Y., Zou, X., Wang, C., Zhang, P., Feng, W., 2016. Robust grape cluster detection in a vineyard by combining the adaboost framework and multiple color components. Sensors (Switzerland) 16 (12), 1–20. https://doi.org/10.3390/s16122098.
Mao, S., Li, Y., Ma, Y., Zhang, B., Zhou, J., Wang, Kai, 2020. Automatic cucumber recognition algorithm for harvesting robots in the natural environment using deep learning and multi-feature fusion. Comput. Electron. Agric. 170 (December 2019) https://doi.org/10.1016/j.compag.2020.105254.
Massah, J., Asefpour Vakilian, K., Shabanian, M., Shariatmadari, S.M., 2021. Design, development, and performance evaluation of a robot for yield estimation of kiwifruit. Comput. Electron. Agric. 185 (January), 106132. https://doi.org/10.1016/j.compag.2021.106132.
Mavridou, E., Vrochidou, E., Papakostas, G.A., Pachidis, T., Kaburlasos, V.G., 2019. Machine vision systems in precision agriculture for crop farming. J. Imag. 5 (12) https://doi.org/10.3390/jimaging5120089.
Mazzia, V., Khaliq, A., Salvetti, F., Chiaberge, M., 2020. Real-time apple detection system using embedded systems with hardware accelerators: an edge AI application. IEEE Access 8, 9102–9114. https://doi.org/10.1109/Access.628763910.1109/ACCESS.2020.2964608.
Mehta, S.S., Ton, C., Asundi, S., Burks, T.F., 2017. Multiple camera fruit localization using a particle filter. Comput. Electron. Agric. 142, 139–154. https://doi.org/10.1016/j.compag.2017.08.007.
Méndez, V., Pérez-Romero, A., Sola-Guirado, R., Miranda-Fuentes, A., Manzano-Agugliaro, F., Zapata-Sierra, A., Rodríguez-Lizana, A., 2019. In-field estimation of orange number and size by 3D laser scanning. Agronomy 9 (12), 885. https://doi.org/10.3390/agronomy9120885.
Nasir, I.M., Bibi, A., Shah, J.H., Khan, M.A., Sharif, M., Iqbal, K., Nam, Y., Kadry, S., 2020. Deep learning-based classification of fruit diseases: an application for precision agriculture. Comput. Mater. Continua 66 (2), 1949–1962. https://doi.org/10.32604/cmc.2020.012945.
NIVIDA, 2021. NVIDIA Isaac Sim. https://developer.nvidia.com/isaac-sim.
Okamoto, H., Lee, W.S., 2009. Green citrus detection using hyperspectral imaging. Comput. Electron. Agric. 66 (2), 201–208. https://doi.org/10.1016/j.compag.2009.02.004.
Oliveira, L.F.P., Moreira, A.P., Silva, M.F., 2021. Advances in agriculture robotics: a state-of-the-art review and challenges ahead. Robotics 10 (2), 1–31. https://doi.org/10.3390/robotics10020052.
Onishi, Y., Yoshida, T., Kurita, H., Fukao, T., Arihara, H., Iwai, A., 2019. An automated fruit harvesting robot by using deep learning. ROBOMECH J. 6 (1), 2–9. https://doi.org/10.1186/s40648-019-0141-2.
Pourdarbani, R., Sabzi, S., Hernández-Hernández, M., Hernández-Hernández, J.L., García-Mateos, G., Kalantari, D., Molina-Martínez, J.M., 2019. Comparison of different classifiers and the majority voting rule for the detection of plum fruits in garden conditions. Remote Sens. 11 (21), 1–17. https://doi.org/10.3390/rs11212546.
Raj, T., Hashim, F.H., Huddin, A.B., Ibrahim, M.F., Hussain, A., 2020. A survey on LiDAR scanning mechanisms. Electronics (Switzerland) 9 (5), 741. https://doi.org/10.3390/electronics9050741.
Rehman, T.U., Mahmud, M.S., Chang, Y.K., Jin, J., Shin, J., 2019. Current and future applications of statistical machine learning algorithms for agricultural machine vision systems. Comput. Electron. Agric. 156, 585–605. https://doi.org/10.1016/j.compag.2018.12.006.
Rodríguez, A., 2021. Smart Farming, ¿qué es y cuál es su futuro? Telcel Tendencias. https://www.telcel.com/empresas/tendencias/notas/que-es-smart-farming.html.
Rosales-Soto, A., Arechavala-Vargas, R., 2020. Agricultura inteligente en México : Analítica de datos como herramienta de competitividad. Vinculategia 1415–1427.
Roy, S.K., De, D., 2020. Genetic algorithm based internet of precision agricultural things (IopaT) for agriculture 4.0. Int. Things 100201. https://doi.org/10.1016/j.iot.2020.100201.
Sa, I., Lehnert, C., English, A., McCool, C., Dayoub, F., Upcroft, B., Perez, T., 2017. Peduncle detection of sweet pepper for autonomous crop harvesting-combined color and 3-D information. IEEE Rob. Autom. Lett. 2 (2), 765–772. https://doi.org/10.1109/LRA.2017.2651952.
Sepulveda, D., Fernandez, R., Navas, E., Armada, M., Gonzalez-De-Santos, P., 2020. Robotic aubergine harvesting using dual-arm manipulation. IEEE Access 8, 121889–121904. https://doi.org/10.1109/Access.628763910.1109/ACCESS.2020.3006919.
Slaughter, D.C., Harrell, R.C., 1989. Discriminating fruit for robotic harvest using color in natural outdoor scenes. Trans. Am. Soc. Agric. Eng. 32 (2), 757–763.
Su, D., Kong, H., Qiao, Y., Sukkarieh, S., 2021. Data augmentation for deep learning based semantic segmentation and crop-weed classification in agricultural robotics. Comput. Electron. Agric. 190 (August), 106418. https://doi.org/10.1016/j.compag.2021.106418.
Tang, Y., Chen, M., Wang, C., Luo, L., Li, J., Lian, G., Zou, X., 2020. Recognition and localization methods for vision-based fruit picking robots: a review. Front. Plant Sci. 11 (May), 1–17. https://doi.org/10.3389/fpls.2020.00510.
Tsoulias, N., Paraforos, D.S., Xanthopoulos, G., Zude-Sasse, M., 2020. Apple shape detection based on geometric and radiometric features using a LiDAR laser scanner. Remote Sens. 12 (15), 2481. https://doi.org/10.3390/rs12152481.
Tu, S., Pang, J., Liu, H., Zhuang, N., Chen, Y., Zheng, C., Wan, H., Xue, Y., 2020. Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images. Precis. Agric. 21 (5), 1072–1091. https://doi.org/10.1007/s11119-020-09709-3.
Vasconez, J.P., Admoni, H., Auat Cheein, F., 2021. A methodology for semantic action recognition based on pose and human-object interaction in avocado harvesting processes. Comput. Electron. Agric. 184 (December 2020) https://doi.org/10.1016/j.compag.2021.106057.
Wang, C., Tang, Y., Zou, X., Luo, L., Chen, X., 2017a. Recognition and matching of clustered mature litchi fruits using binocular charge-coupled device (CCD) color cameras. Sensors (Switzerland) 17 (11), 2564. https://doi.org/10.3390/s17112564.
Wang, L.L., Zhao, B., Fan, J.W., Hu, X.A., Wei, S., Li, Y.S., Zhou, Q.B., Wei, C.F., 2017b. Development of a tomato harvesting robot used in greenhouse. Int. J. Agric. Biol. Eng. 10 (4), 140–149. https://doi.org/10.25165/j.ijabe.20171004.3204.
Wang, T., Xu, X., Wang, C., Li, Z., Li, D., 2021. From smart farming towards unmanned farms: a new mode of agricultural production. Agriculture (Switzerland) 11 (2), 1–26. https://doi.org/10.3390/agriculture11020145.
Whittaker, A.D., Miles, G.E., Mitchell, O.R., Gaultney, L.D., 1987. Fruit location in a partially occluded image. Trans. Am. Soc. Agric. Eng. 30 (3), 591–596. https://doi.org/10.13031/2013.30444.
Williams, H., Ting, C., Nejati, M., Jones, M.H., Penhall, N., Lim, J., Seabright, M., Bell, J., Ahn, H.S., Scarfe, A., Duke, M., MacDonald, B., 2020. Improvements to and large-scale evaluation of a robotic kiwifruit harvester. J. Field Rob. 37 (2), 187–201. https://doi.org/10.1002/rob.v37.210.1002/rob.21890.
Wu, G., Li, B., Zhu, Q., Huang, M., Guo, Y., 2020. Using color and 3D geometry features to segment fruit point cloud and improve fruit recognition accuracy. Comput. Electron. Agric. 174 (May), 105475. https://doi.org/10.1016/j.compag.2020.105475.
Xiang, Z., Chen, K., Qian, M., Hu, X., 2020. Yarn-dyed woven fabric density measurement method and system based on multi-directional illumination image fusion enhancement technology. J. Text. Inst. 111 (10), 1489–1501. https://doi.org/10.1080/00405000.2019.1706222.
Xiao, Z., Qi, J., Xue, W., Zhong, P., 2021. Few-shot object detection with self-adaptive attention network for remote sensing images. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 14, 4854–4865. https://doi.org/10.1109/JSTARS.2021.3078177.
Xiong, Y.a., Ge, Y., Grimstad, L., From, P.J., 2019a. An autonomous strawberry-harvesting robot: design, development, integration, and field evaluation. J. Field Rob. 37 (2), 202–224. https://doi.org/10.1002/rob.v37.210.1002/rob.21889.
Xiong, Y., Peng, C., Grimstad, L., From, P.J., Isler, V., 2019b. Development and field evaluation of a strawberry harvesting robot with a cable-driven gripper. Comput. Electron. Agric. 157 (December 2018), 392–402. https://doi.org/10.1016/j.compag.2019.01.009.
Yan, B., Fan, P., Lei, X., Liu, Z., Yang, F., 2021. A real-time apple targets detection method for picking robot based on improved YOLOv5. Remote Sens. 13 (9), 1–23. https://doi.org/10.3390/rs13091619.
Yang, C.H., Xiong, L.Y., Wang, Z., Wang, Y., Shi, G., Kuremot, T., Zhao, W.H., Yang, Y., 2020a. Integrated detection of citrus fruits and branches using a convolutional neural network. Comput. Electron. Agric. 174 (May), 105469. https://doi.org/10.1016/j.compag.2020.105469.
Yang, Q., Chen, C., Dai, J., Xun, Y., Bao, G., 2020b. Tracking and recognition algorithm for a robot harvesting oscillating apples. Int. J. Agric. Biol. Eng. 13 (5), 163–170. https://doi.org/10.25165/j.ijabe.20201305.5520.
Yasmin, J., Lohumi, S., Raju Ahmed, M., Mohan Kandpal, L., Akbar Faqeerzada, M., Sung Kim, M., Cho, B.-K., 2020. Improvement in purity of healthy tomato seeds using an image-based one-class classification method. Sensors (Switzerland) 20 (2690).
Yost, M.A., Kitchen, N.R., Sudduth, K.A., Sadler, E.J., Drummond, S.T., Volkmann, M.R., 2017. Long-term impact of a precision agriculture system on grain crop production. Precis. Agric. 18 (5), 823–842. https://doi.org/10.1007/s11119-016-9490-5.

Zeng, L., Feng, J., He, L., 2020. Semantic segmentation of sparse 3D point cloud based on geometrical features for trellis-structured apple orchard. Biosyst. Eng. 196, 46–55. https://doi.org/10.1016/j.biosystemseng.2020.05.015.

Zhang, H., Zhang, S., Dong, W., Luo, W., Huang, Y., Zhan, B., Liu, X., 2020a. Detection of common defects on mandarins by using visible and near infrared hyperspectral imaging. Infrared Phys. Technol. 108 (April), 103341. https://doi.org/10.1016/j.infrared.2020.103341.

Zhang, T., Huang, Z., You, W., Lin, J., Tang, X., Huang, H., 2020b. An autonomous fruit and vegetable harvester with a low-cost gripper using a 3D sensor. Sensors (Switzerland) 20 (1), 93. https://doi.org/10.3390/s20010093.

Zhang, Y.-M., Lee, C.-C., Hsieh, J.-W., Fan, K.-C., 2021. CSL-YOLO: A New Lightweight Object Detection System for Edge Computing, pp. 1–12. http://arxiv.org/abs/2107.04829.

Zhao, Y., Gong, L., Huang, Y., Liu, C., 2016. A review of key techniques of vision-based control for harvesting robot. Comput. Electron. Agric. 127, 311–323. https://doi.org/10.1016/j.compag.2016.06.022.

Zhou, C., Hu, J., Xu, Z., Yue, J., Ye, H., Yang, G., 2020. A novel greenhouse-based system for the detection and plumpness assessment of strawberry using an improved deep learning technique. Front. Plant Sci. 11 (June), 1–13. https://doi.org/10.3389/fpls.2020.00559.

Zhuang, C., Wang, Z., Zhao, H., Ding, H., 2021. Semantic part segmentation method based 3D object pose estimation with RGB-D images for bin-picking. Rob. Comput. Integr. Manuf. 68, 102086. https://doi.org/10.1016/j.rcim.2020.102086.

Zoto, J., Musci, M.A., Khaliq, A., Chiaberge, M., Aicardi, I., 2020. Automatic path planning for unmanned ground vehicle using UAV imagery. In: Advances in Intelligent Systems and Computing, Vol. 980. Springer International Publishing. https://doi.org/10.1007/978-3-030-19648-6_26.