agriculture

MDPI

Review

A Crawling Review of Fruit Tree Image Segmentation

Il-Seok Oh [1] and Jin-Seon Lee [2,*]

[1] Department of Computer Science and Artificial Intelligence/CAIIT, Jeonbuk National University, Jeonju 54896, Republic of Korea; isoh@jbnu.ac.kr
[2] Department of Information Security, Woosuk University, Wanju 55338, Republic of Korea
* Correspondence: jslee@woosuk.ac.kr

Abstract

Fruit tree image segmentation is an essential problem in automating a variety of agricultural tasks such as phenotyping, harvesting, spraying, and pruning. Many research papers have proposed a diverse spectrum of solutions suitable for specific tasks and environments. The review scope of this paper is confined to the front views of fruit trees, and 207 relevant papers proposing tree image segmentation in an orchard environment are collected using a newly designed crawling review method. These papers are systematically reviewed based on a four-tier taxonomy that sequentially considers the method, image, task, and fruit. This taxonomy will assist readers to intuitively grasp the big picture of these research activities. Our review reveals that the most noticeable deficiency of the previous studies was the lack of a versatile dataset and segmentation model that could be applied to a variety of tasks and environments. Six important future research topics, such as building large-scale datasets and constructing foundation models, are suggested, with the expectation that these will pave the way to building a versatile tree segmentation module.

Keywords: agricultural task; precision farming; rule-based method; deep learning; computer vision; crawling review

check for updates

Academic Editors: Alessandra Vinci and Daniela Farinelli

Received: 16 September 2025
Revised: 22 October 2025
Accepted: 24 October 2025
Published: 27 October 2025

Citation: Oh, I.-S.; Lee, J.-S. A Crawling Review of Fruit Tree Image Segmentation. *Agriculture* **2025**, *15*, 2239. https://doi.org/10.3390/agriculture15212239

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

To maximize the production of fruit trees in orchards, farmers and horticulturists perform various tasks such as phenotyping, health and growth monitoring, spraying, pruning, and yield estimation. In the past, these tasks were performed manually. However, the fatigue of the human laborers often led to imprecise and error-prone results. In addition, the continual increase in labor costs has resulted in high market prices for fruits. The automation of these tasks has become very important and garnered much attention from agronomists and computer scientists [1–3].

Computer vision researchers have continually attempted to develop automatic systems for these agricultural tasks [4,5]. However, the large variations in the tree arrangements in orchards, tree shapes, weather, and image acquisition devices have made traditional computer vision algorithms far from practical. Recently, owing to the advent of deep learning technology, practical systems are possible, and commercial systems are available. Deep learning has transformed the fundamentals of computer vision from manual design through human reasoning to machine learning-based design through neural network optimization [6].

Obtaining relevant information about individual trees through detailed observations of their trunks, branches, leaves, and fruit is essential to successfully accomplish the necessary tasks. Therefore, an important preliminary problem in automating these tasks is tree image

segmentation [7]. Segmenting the regions containing individual trees is required. Often a finer segmentation of the tree region into the trunk, branches, leaves, and fruit must also be performed. The tree segmentation problem is challenging because of several factors. First, the acquired images have large variations. Many factors influence these variations, including the fruit type, geography, weather, and farming techniques. Second, occlusion by various facilities such as trellises and poles commonly occurs. The self occlusion produced by leaves and branches makes the segmentation problem difficult. Third, because fruit trees are placed in a row and separated by equal distances, their branches are commonly intertwined. The boundaries between adjacent trees are ambiguous.

Recently, a survey paper on tree image segmentation was published [7]. Although the paper considered images from orchards and a forest, it gave more attention to the top-view images acquired by unmanned aerial vehicles (UAVs). Therefore, the application tasks made possible by the segmentation results were oriented toward the digital forestry domain. In contrast, agricultural tasks such as harvesting, spraying, and pruning require the segmentation of the front-view images of trees. To the best of our knowledge, there has been no survey or review paper dealing with fruit tree segmentation in the agricultural domain. This fact motivated our study.

The aim of this paper is to present a review of fruit tree image segmentation algorithms, involving both traditional rule-based (RB) and modern deep learning (DL) approaches. The scope of this review is confined to segmenting front-view images of fruit trees. It excludes the segmentation of forest trees, which belongs to the digital forestry domain. Using a newly designed crawling review method, 207 relevant papers were collected. This review was performed by grouping the papers according to several criteria. The hierarchy of the taxonomy had the following order: the technical approach (RB vs. DL), image type (RGB, RGB-D, point cloud, others), task (harvesting, phenotyping, spraying, pruning, yield estimation, etc.), fruit type (apple, grape, citrus, pear, peach, litchi, guava, cherry, etc.), and publication year (from old to recent).

The main contributions of this review are summarized as follows:

1. Focus on frontal-view image segmentation in orchard environments: Unlike most existing reviews that concentrate on top-view images captured by UAVs or satellites, this work specifically targets segmentation of frontal-view images of fruit trees within orchard environments.
2. Coverage of recent literature: The review includes an extensive collection of papers published up to the end of 2025, ensuring that the most recent advancements are captured.
3. Introduction of a new review methodology: A novel review approach, termed *crawling review*, is proposed and applied in the paper to enhance the comprehensiveness and systematic nature of the literature search.
4. Proposal of a four-tier taxonomy: A hierarchical taxonomy is developed and used to classify the reviewed studies across four levels—methodology, image type, agricultural task, and fruit type—providing a structured and insightful overview of the field.

The remainder of this paper is organized as follows: Section 2 outlines this review, focusing on the review scope, method, and statistics. Section 3 reviews the papers that reported results based on an RB approach. Section 4 reviews the papers that reported results based on DL. Section 5 presents a discussion and future work. Section 6 concludes the paper.

2. Review Scope, Method, and Statistics

Because tree segmentation is a broad research topic, this paper focuses on the agricultural domain. Section 2.1 describes the review scope in detail. Section 2.2 explains how this

paper collected and filtered the related papers, along with the taxonomy used. Section 2.3 gives some statistics about the papers.

2.1. Review Scope

Tree segmentation can be divided into two domains: fruit trees in an agricultural domain and forest trees in a digital forestry domain. Fruit trees are usually planted in orchards in rows. The typical tasks supported by tree segmentation in this domain are related to precision farming and include phenotyping, harvesting, spraying, pruning, yield estimation, and robot navigation. These tasks require the segmenting of individual trees and often the finer segmentation of a tree region into parts such as the trunk, branches, leaves, and fruit. The digital forestry domain treats a larger area where the trees are randomly or regularly planted. It requires a coarse segmentation of trees covering a large forest. Typical tasks include species and habitat identification, population estimation, spraying, health and growth monitoring, and logging planning.

Two different methods are used to capture tree images. In the first, a person or robot captures images in front of target trees, resulting in front-view images. In the second, top-view images are acquired using unmanned aerial vehicles (UAVs) carrying image sensors. Front-view images are the major type utilized in the agricultural domain, while top-view images are primarily used in the digital forestry domain. Figure 1 illustrates examples of front- and top-view tree images.



Figure 1. Tree images: front view of apple trees on **left** [8] and top view of cherry trees on **right** [9].

This review is confined to the front-view images of fruit trees. Chronologically, the review covers a long period extending from 1990 to 2025. Methodologically, it includes both RB and DL approaches. The RB algorithms include thresholding, clustering, region growing, machine learning, fitting, and graph-based methods. Papers reporting the results of studies that adopted DL began to appear in the late 2010s. These DL algorithms used convolutional neural network (CNN) models such as mask R-CNN and YOLACT, and transformer models such as DETR and Swin. Appendices A.1.2 and A.1.3 briefly explain these algorithms.

This paper excludes papers that deal only with the segmentation of the fruit, which is regarded as a separate problem. Readers are referred to another survey paper [10] for these fruit-only segmentation algorithms. Papers that discuss the segmenting of the trunk only, branches only, or flowers only are included. Moreover, papers that discuss the segmenting of the fruit along with the fruit-bearing stems are included. Figure 2 demonstrates several levels of tree segmentation, including whole tree, branch, branch classification, fruit with stem, fruit with picking point, and flower segmentation.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 2. Various levels of tree segmentation: (**a**) whole tree [8], (**b**) branch [11], (**c**) branch classification [12], (**d**) fruit and stem [13], (**e**) fruit and picking point [14], and (**f**) flower [15]).

The explanation of the papers in this review focuses on the methodology rather than the performance or effect. Because of the lack of a standard dataset, an objective performance comparison between different methods is currently of little importance.

2.2. Literature Search and Taxonomy

2.2.1. Crawling Review

A *systematic review* is the standard method used by most review papers. In a systematic review, the authors collect a preliminary set of relevant papers by searching databases such as the Web of Science (WoS) or Google scholar using a well-designed query [16]. This preliminary set is then filtered by the authors to select the final set of papers, which is the reading and review target. For example, in one review paper [7] on tree segmentation, the authors searched Google scholar for papers confined to the period of 2013–2023 using the query keywords "UAV," "tree," "segmentation," and "classification." They collected 979 peer-reviewed papers as a preliminary set and filtered them to obtain 144 papers as their final set.

Because of the high variability of fruit tree segmentation, as demonstrated in Figure 2, it is assumed that the review required another search method for relevant papers. The lack of a standard dataset and insufficiency of review or survey papers on tree segmentation support this argument. This paper proposed a novel review method, called a *crawling review*. The search process of a crawling review is similar to that of web crawling. Just as web crawling starts with a queue containing a seed URL, a crawling review starts with a queue Q containing a seed paper or set of seed papers. Popping the most recent paper p from Q, the relevant papers cited in p are then pushed into the queue. This process of article collection by popping a paper from Q and adding relevant papers into Q continues until Q is empty. Figure 3a explains this process formally. Figure 3b shows the supplementary phase, which collects additional articles from the most recent issues of the most cited journals. This supplementary phase is optional.

In this review, ref. [7] was the seed paper used to initiate the queue, Q. To our knowledge, the paper provides the most comprehensive and in-depth treatment of the topic to date. We utilized the supplementary phase, where the J queue was initiated with four journals: *Computers and Electronics in Agriculture*, *Biosystems Engineering*, *Agriculture*, and *Journal of Field Robotics*. The given period was set to January 2020–August 2025.

The quality of the conventional review method using a well-designed query is heavily dependent on the query design and the search engine. We believe that this novel crawling review will result in better quality because it is closer to an exhaustive search. However, the crawling review is much more laborious and takes longer because the search process is performed manually. Additionally, it lacks reproducibility due to the manual selection.

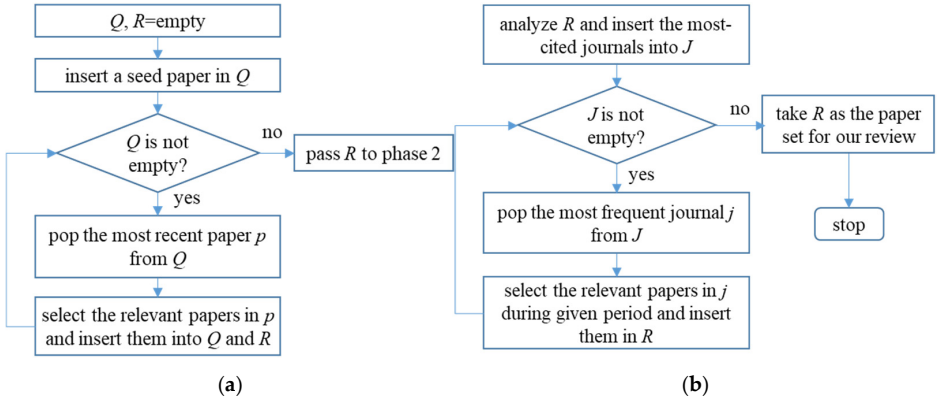


Figure 3. Crawling review with supplementary phase. (**a**) Phase 1: crawling (**b**) Phase 2: supplementing.

2.2.2. Four-Tier Taxonomy

The papers collected by the process illustrated in Figure 3 are classified according to taxonomy in Figure 4. The classification hierarchy has the following order: the tree type (fruit or forest), view (front or top), method, image type, task, and fruit type. As mentioned in the previous section, the forest tree and top-view images were excluded from this review. Another classification criterion, not illustrated in Figure 4, was the fruit type (e.g., apple, grape, citrus, peach, and guava). The fruit types are listed in Tables 1 and 2, where the relevant papers are listed. The tables are organized according to the taxonomy presented in Figure 4, which includes the method, image type, task, publication year, and fruit type. The publication year is included to help readers follow the chronological development of the reviewed studies.

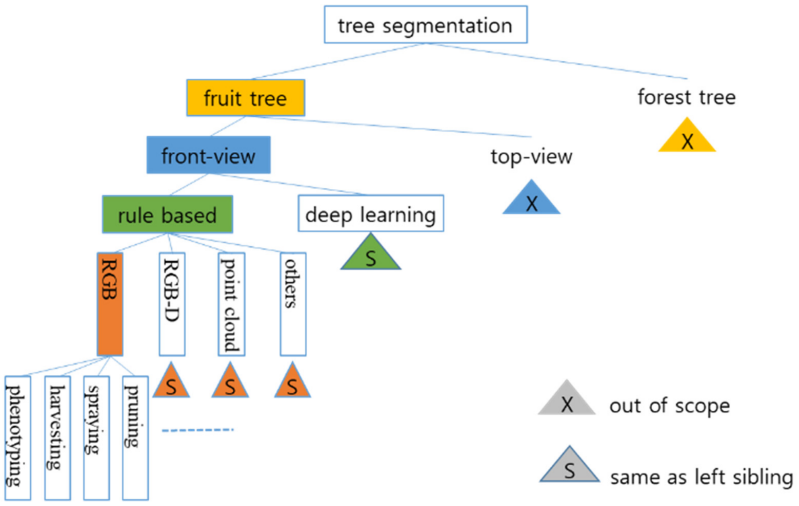


Figure 4. Taxonomy of literature and review scope.

Table 1. Summary of RB tree image segmentation papers.

Image Type	Agricultural Tasks and Relevant Papers
RGB	**Phenotyping**: [17] (2017, apple), [18] (2018, apple), [19] (2002, grape), [20] (2025, peach) **Harvesting**: [21] (2016, apple), [22] (2017, apple), [23] (2017, apple), [24] (2018, tomato), [25] (2011, litchi), [26] (2019, litchi), [27] (2018, litchi), [28] (2018, litchi), [29] (1993, citrus), [30] (2014, citrus), [31] (2018, citrus), [32] (2016, cherry), [33] (2017, cherry), [34] (2023, palm), [35] (2012, Chinese hickory), [36] (2014, Chinese hickory) **Spraying**: [37] (2010, apple), [38] (2010, grape), [39] (2019, olive), [40] (2021, cherry) **Pruning**: [41] (1997, grape), [42] (2006, grape) **Yield estimation**: [43] (2018, apple), [44] (2024, apple) **Navigation**: [45] (2016, palm) **Thinning**: [46] (2022, apple)

Table 1. *Cont.*

Image Type	Agricultural Tasks and Relevant Papers
RGB-D	**Phenotyping**: [47] (2018, pear) **Harvesting**: [48] (2020, guava, pepper, eggplant) **Spraying**: [49] (2017, grape, peach, apricot), [50] (2018, grape), [51] (2020, peach) **Navigation**: [52] (2022, pear)
Point cloud	**Phenotyping**: [53] (2009, apple, pear, grape), [54] (2009, apple, pear, grape, citrus), [55] (2014, apple, pear, grape), [56] (2015, apple, grape), [57] (2016, apple), [58] (2020, apple), [59] (2015, grape), [60] (2015, grape), [61] (2017, grape), [62] (2023, tomato), [63] (2023, tomato), [64] (2015, pear), [65] (2012, peach), [66] (2015, almond), [67] (2015, apple), [68] (2016, almond), [69] (2021, mango, avocado), [70] (2025, apple) **Harvesting**: [71] (2017, apple) **Spraying**: [72] (2021, apple), [73] (2025, apple) **Pruning**: [74] (2014, apple), [75] (2015, apple), [76] (2020, apple), [77] (2022, cherry), [78] (2022, cherry), [79] (2023, jujube), [80] (2021, mango, avocado), [81] (2025, apple, cherry), [82] (2025, cherry) **Yield estimation**: [83] (2021, apple), [84] (2012, grape) **Navigation**: [85] (2013, Not Available) **Thinning**: [86] (2012, peach)
Others	**Phenotyping**: [87] (2012, apple, yucca) **Harvesting**: [88] (2022, grape), [89] (2011, citrus), [90] (2008, cherry), [91] (2013, pepper), [92] (2014, pepper), [93] (2018, olive) **Pruning**: [94] (2016, apple), [95] (2016, apple), [96] (2017, apple), [97] (2013, grape), [98] (2017, grape), [99] (2016, grape)

Table 2. Summary of DL-based tree image segmentation papers.

Image Type	Agricultural Tasks and Relevant Papers
RGB	**Phenotyping**: [100] (2020, apple), [101] (2022, apple), [102] (2022, apple), [103] (2023, apple), [15] (2018, apple, peach, pear), [104] (2022, apple, peach, pear), [105] (2023, citrus), [106] (2024, apple), [107] (2024, cherry), [108] (2024, apple), [109] (2025, apple), [110] (2025, tomato), [111] (2025, apple), [112] (2025, apple), **Harvesting**: [113] (2019, apple), [114] (2020, apple), [11] (2023, apple), [115] (2023, apple), [116] (2023, apple), [117] (2020, grape), [118] (2021, grape), [119] (2022, jujube), [120] (2023, grape), [121] (2023, grape), [122] (2022, tomato), [123] (2022, tomato), [124] (2023, tomato), [125] (2023, tomato), [126] (2020, litchi), [127] (2021, litchi, passion fruit, citrus, guava, jujube), [128] (2021, litchi), [129] (2023, litchi), [130] (2022, guava), [131] (2023, Sichuan pepper), [132] (2023, jujube), [133] (2019, kiwi), [134] (2020, kiwi), [135] (2021, kiwi), [14] (2021, mango), [136] (2022, banana), [137] (2022, pomegranate), [138] (2022, longan), [139] (2024, grape), [140] (2024, tomato), [141] (2024, mango), [142] (2024, mango), [143] (2024, mango), [144] (2024, apple), [145] (2024, grape), [146] (2024, litchi), [147] (2025, kiwi), [148] (2025, grape), [149] (2025, grape), [150] (2025, grape) **Spraying**: [151] (2020, pear), [152] (2022, pear) **Pruning**: [153] (2022, apple), [154] (2023, apple), [155] (2023, grape), [156] (2023, grape), [12] (2023, cherry), [157] (2025, tomato) **Yield estimation**: [158] (2020, apple), [159] (2022, apple), [160] (2023, apple), [8] (2023, apple), [161] (2022, grape) **Navigation**: [162] (2023, pear, peach), [163] (2025, grape) **Thinning**: [13] (2023, apple), [164] (2020, grape), [165] (2020, grape), [166] (2022, banana), [167] (2025, grape), [168] (2025, apple)
RGB-D	**Phenotyping**: [169] (2020, apple), [170] (2019, grape), [171] (2019, cherry), [172] (2018, Capsicum annuum (pepper)), [173] (2019, Capsicum annuum (pepper)), [174] (2020, Capsicum annuum (pepper)), [175] (2024, apple), [176] (2024, tomato) **Harvesting**: [177] (2018, apple), [178] (2019, apple), [179] (2020, apple), [180] (2020, apple), [181] (2022, apple), [182] (2023, grape), [183] (2022, cherry tomato), [184] (2022, tomato), [185] (2023, tomato), [186] (2020, litchi), [187] (2020, citrus), [188] (2019, guava), [189] (2021, guava), [190] (2021, guava), [191] (2022, pomegranate), [192] (2024, tomato), [193] (2024, tomato), [194] (2025, tomato) **Spraying**: [195] (2023, citrus), [196] (2025, apple) **Pruning**: [197] (2021, apple), [198] (2025, apple) **Training**: [199] (2020, apple) **Navigation**: [200] (2024, apple), [201] (2024, peach) **Thinning**: [202] (2025, apple)
Point cloud	**Phenotyping**: [203] (2023, apple), [204] (2021, tomato, maze), [205] (2024, apple), [206] (2024, apple), [207] (2025, apple) **Harvesting**: [208] (2022, grape) **Pruning**: [209] (2021, jujube), [210] (2024, apple), [211] (2024, cherry), [212] (2025, grape), [213] (2025, jujube), [214] (2025, walnut)
Others	**Phenotyping**: [215] (2022, apple), [216] (2023, tomato) **Yield estimation**: [217] (2013, almond)

2.3. Statistics

Visual summaries in Figure 5 are derived from Tables 1 and 2. These statistics capture trends in both methodological approaches and data types over time. Figure 5a shows the annual number of publications related to tree segmentation. A notable increase in total publications is observed from 2010 onward. Rule-based (RB) methods dominated until

around 2017, reaching a peak between 2015 and 2018. In contrast, deep learning (DL) methods were virtually absent before 2018, after which they began to appear and have shown steady growth since. This shift suggests a gradual methodological transition from RB to DL approaches, reflecting broader trends in computer vision and agricultural automation. Figure 5b presents the distribution of data types used across studies. During the RB-dominated period, RGB images and point cloud data were used with similar frequency. RGB imagery is commonly used due to its low cost and wide availability, while point clouds offer robustness to varying illumination conditions. In recent DL-based studies, RGB data has become the predominant input modality. This may be attributed to the ability of DL models to achieve high performance using RGB data alone, reducing the reliance on more complex or expensive data types such as point clouds or multispectral imagery. Overall, the trends in Figure 5 indicate two key developments: (1) a methodological evolution from RB to DL, with DL gaining momentum post-2018, and (2) a growing preference for RGB data, particularly in DL applications, likely due to a balance of performance and cost-efficiency.

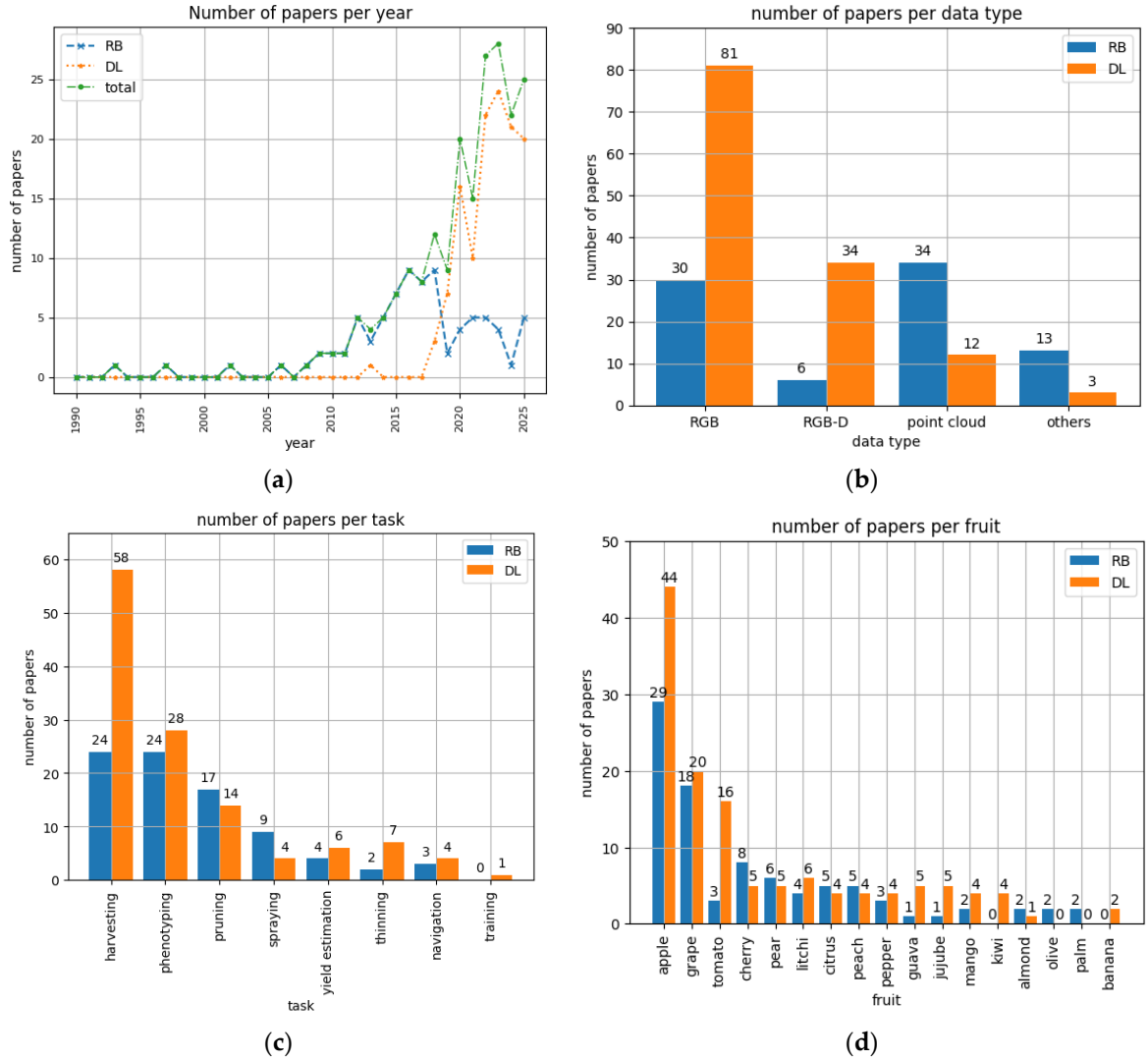


Figure 5. (**a**) Number of papers per year. (**b**) Number of papers per data type. (**c**) Number of papers per task. (**d**) Number of papers per fruit. Various statistics derived from the review results, presented in Tables 1 and 2.

Figure 5c illustrates the number of papers per agricultural task supported by tree segmentation. Phenotyping and harvesting are the two most popular tasks. One noticeable

fact is that in the DL era, harvesting has the highest frequency. This is because DL makes practical and commercial systems possible, and the most urgent task is harvesting because of the high cost of manual harvesting [218]. Figure 5d shows the statistics according to the fruit type. Apples and grapes are two most popular fruits. Additional fruit types, not illustrated in the figure, include avocado, apricot, Chinese hickory, yucca, pomegranate, longan, walnut, and passion fruit.

3. Rule-Based Algorithms for Tree Image Segmentation

Table 1 summarizes the 83 papers that discuss the use of RB segmentation algorithms. It follows the taxonomy hierarchy shown in Figures 4 and 6. The rows are ordered in accordance with the hierarchy level of the image type. In each row, the papers are listed in terms of agricultural tasks, in the order shown in Figure 6a. For each task, the papers are further listed in order of the fruit type, as shown in Figure 6b. Each paper shows the fruit type in parentheses. Some papers consider multiple fruit types. The section structure is strictly in accordance with Table 1. The papers discussing the same fruit type are placed in the same paragraph. When a research group has published a series of papers, the papers are reviewed together to make it more convenient to track the research progress.

(a) harvesting → phenotyping → pruning → spraying → yield estimation → thinning → navigation → training
(b) apple → grape → tomato → cherry → pear → litchi → citrus → peach → pepper → guava → jujube → mango → kiwi → almond → olive → palm → banana → walnut

Figure 6. Order of review: (**a**) in terms of task and (**b**) in terms of fruit.

3.1. RGB

Many studies used RGB images because they are the most popular and cheapest. They used various RB segmentation algorithms, as explained in Appendix A.1.2.

3.1.1. Phenotyping

Tabb et al. measured tree traits such as the structure, diameter, length, and angle of the branches of apple trees for the phenotyping and pruning tasks [17]. They captured tree images by placing a blue curtain behind the tree to ease the processing of the vision algorithm. A six-stage procedure was applied, which included calibration, segmentation, reconstruction, skeletonization, graph representation, and feature of interest extraction. The same research group improved their work by adopting super-pixel and GMM methods [18].

Svensson et al. proposed a leaf and trunk region segmentation algorithm for the tasks of shoot counting and canopy assessment of grapevines [19]. They placed curtains of various colors and evaluated the effectiveness of those colors. The 3D Gaussian splatting (3DGS) algorithm was enhanced to be suitable for 3D reconstruction of peach trees from the multi-view RGB images [20]. The distance-weighted filtering and super-sampling techniques were integrated into 3DGS.

3.1.2. Harvesting

With the aim of building an apple picking robot, Ji et al. converted the color space from RGB to XYZI1I2I3 [21]. By analyzing the proper threshold ranges for the fruit, branches, leaves, and sky, an image was segmented into different regions. The same research group proposed another method that used mean shift clustering to identify the fruit, leaves, and branches [22]. For apple picking, Silwal et al. captured apple tree images by placing a black curtain behind a tree [23]. To alleviate the illumination variation, five images were combined using the exposure fusion technique. Xiang et al. proposed an algorithm for segmenting tomato plant images [24]. Their method used images captured at night to

reduce the illumination variation during the daytime. The grayscale image was segmented using a pulse-coupled neural network (PCNN).

Deng et al. proposed an algorithm that segmented a litchi tree image into fruit-bearing branches [25]. They obtained string-like litchi regions by thresholding the Cr channel of the YCbCr color space. The litchi fruits were identified by applying k-means clustering to the string-like litchi regions. Zhuang et al. proposed an algorithm for picking litchi [26]. An iterative retinex algorithm was applied to reduce the illumination effect while keeping the chromatic information. Ripe litchi fruit regions were segmented by applying Otsu thresholding to RG chromatic mapping images. Xiong et al. proposed a litchi tree segmentation algorithm and picking point determination method [27]. It used images captured at night to alleviate the illumination effect. In the YIQ color space, fuzzy clustering method was applied to remove the background. Then, using Otsu thresholding, the stem and fruit regions were segmented. The same research group extended their work by using binocular stereo to measure the distance to the target fruit [28].

Pla et al. proposed a citrus tree segmentation method that identified the fruit, leaf, and sky regions [29]. To diminish the illumination and highlight effects, they constructed a 2D directional space using a dichromatic reflection model and applied a clustering algorithm in this space. Qiang et al. proposed a fruit and branch segmentation algorithm for citrus tree images [30]. Representing a pixel with an (r,g,b) feature vector, a multi-class SVM was trained to classify the pixels into four classes: fruit, branches, leaves, and background. The segmentation result was used for fruit detection and path planning for robot harvesting. Liu et al. presented a segmentation algorithm for harvesting citrus trees [31]. In the training stage, after converting an RGB image to the Y'CbCr color space, multi-elliptical boundary models were constructed in the Cr-Cb space for each fruit and stem region.

With the aim of selecting the shaking point for a shake-and-catch machine, Amatya et al. proposed a segmentation algorithm for cherry tree images [32]. To reduce the illumination effect, the images were captured at night using white LED illumination. Representing a pixel with an (r,g,b) feature vector, a Bayesian model was used to classify the pixels into fruit, branch, leaf, and background regions. The same research group extended their work by using 3D information [33].

Mohammadi et al. proposed a stem and leaf area segmentation method for cutting and picking the date bunches of palm trees [34]. Two smartphones were used to capture RGB images. The image from the first camera was used for stem segmentation by thresholding each of the RGB channels and combining them. The image from the second camera was used to identify the leaf area.

With the aim of building a shake-and-catch harvesting machine, He et al. proposed a 3D branch reconstruction method for Chinese hickory tree images [35]. They used two images taken from different viewpoints. From each image, tree branches were roughly estimated using the snake segmentation algorithm. The branch regions from two images were matched to reconstruct a 3D skeleton of tree branches. The same research group extended their work by presenting an improved segmentation method [36]. A method to determine the optimal vibration frequency for the trunk shaker was also presented.

3.1.3. Spraying

Hocevar et al. proposed an approach for spraying apple trees [37]. The green area was extracted as the tree crown by thresholding each channel of an HSI image, combining the results, and applying the erosion operation. Experiments conducted with water-sensitive papers showed a pesticide reduction of 23%. Berenstein et al. presented a grapevine segmentation algorithm for an automatic spraying task [38]. They presented algorithms for

foliage and grape cluster segmentation. The foliage segmentation was accomplished by analyzing the green channel because the foliage area was green.

Asaei et al. proposed a segmentation method for olive tree images [39]. Thresholding the green channel made it possible to identify the pixels that belonged to the tree region. When the ratio of green pixels exceeded a preset threshold, the nozzle was opened. A method for segmenting cherry tree crowns in a complex background was proposed for building a spraying system [40]. The CRF (conditional random field) and mean field approximation techniques were applied.

3.1.4. Pruning

McFarlane et al. restricted the domain to trained grapevines that had two long branches connected to a trunk [41]. The images were captured during winter when the vines were dormant. The wire was identified using thresholding and the Hough transform. The trunk was identified by analyzing a vertical histogram of the thresholded binary image. Gao et al. used a white curtain behind a grapevine to easily segment branch regions by thresholding [42]. By using a rule set, the canes were located first and then the nodes were found.

3.1.5. Yield Estimation

Roy et al. proposed a 3D reconstruction method for rows of trees in an apple orchard for yield estimation or phenotyping [43]. Each of the frontside and backside images was processed to obtain point clouds. Based on the constraints that the occlusion boundary from two images were the same and trunks were at similar heights, the pair was optimally aligned under an objective function using various techniques such as principal component analysis (PCA), RANSAC, bundle adjustment, and GMM. Aiming to build a smartphone app for yield estimation, Duncan et al. proposed a color ratio-based apple segmentation algorithm [44].

3.1.6. Navigation

With the aim of building an autonomous robot navigation system, Juman et al. proposed a method based on the Viola–Jones algorithm to segment the tree trunk in a palm tree image [45]. The paper modified the Viola–Jones algorithm to improve the trunk segmentation accuracy using depth information.

3.1.7. Thinning

With the aim of flower thinning, Zhang et al. proposed a 3D tree reconstruction method for apple tree rows using three RGB images [46]. Left, right, and top images were acquired using a UAV. Based on PCA, three images were aligned to reconstruct a 3D tree model. The flower regions were segmented using a thresholding technique and mapped to the 3D tree model.

3.2. RGB-D

Because of the availability and low cost of RGB-D cameras after the release of Kinect in 2010, research based on RGB-D images has rapidly increased. A common strategy of methods based on RGB-D images is to enhance the accuracy of the segmentation using additional depth information. The studies differed in how the depth information was used.

3.2.1. Phenotyping

Xue et al. proposed a trunk detection method for pear trees [47]. They acquired RGB images and separately obtained depth images using a SICK LMS 92 laser scanner. The color image was used to segment the trunk of a tree. The trunk width was measured

using the depth image. These two sources of information were fused using the Dempster–Shafer theory.

3.2.2. Harvesting

With the aim of building a picking robot, Lin et al. proposed a segmentation algorithm for guava, pepper, and eggplant images [48]. Kinect v2 was used to acquire RGB-D images. First, the RGB image was segmented using a probabilistic segmentation algorithm that relied on prior probabilities for the foreground (fruit) and background (leaves, branches, soil, and sky). The filtered image obtained by multiplying the segmented binary image and depth image contained both foreground and background regions.

3.2.3. Spraying

Xiao et al. proposed a leaf wall segmentation method for spraying devices [49]. Vineyard, peach, and apricot images acquired with Kinect v1 were used. Assuming that the leaf color was mainly green, the green channel was thresholded to obtain a pre-segmented binary map. The spraying control module calculated the distance between the obtained leaf wall and spraying device to control the amount of pesticide. The same research group improved the system by acquiring front and side images of a vineyard using two Kinect cameras [50]. They combined the two segmentation results to improve the accuracy. The same research group proposed another improved segmentation method that combined the color and depth segmentation maps and applied their method to a peach tree image [51].

3.2.4. Navigation

With the aim of facilitating autonomous navigation along tree rows, Gimenez et al. proposed a trunk segmentation method for RGB-D video captured in a pear orchard [52]. The trunk region was obtained by using a series of processes to discard non-trunk pixels. Assuming that the trees were planted in a regular pattern and the robot had a linear velocity, a simultaneous localization and mapping (SLAM) algorithm was used for the trunk regions to construct a tree row model using the depth map.

3.3. Point Cloud

3.3.1. Phenotyping

Polo et al. proposed a tree segmentation method to extract various properties of apple, pear, and grape trees [53]. The LMS-200 LiDAR sensor was used to acquire the point cloud data of a tree row. Several traits such as volume and surface area were extracted using a numerical algorithm. Rosell et al. proposed a 3D reconstruction method of a tree row [54]. The datasets were constructed for various species of apples, pears, grapes, and citrus fruits using the LMS-200 LiDAR sensor. The data measured at different positions were integrated into a 3D map that represented a tree row. Mendez et al. proposed a tree segmentation method for modeling and maintaining the evolution of apple, pear, and grape orchards [55]. Starting from the bottom point, a set of points were fitted to a cylinder that represented the trunk. Considering the fact that the successor branches were thinner than their parents, a new branch was generated, and a new cylinder was fitted as the successor branch. Das et al. built a sensor suite having a laser range scanner, a multi-spectral camera, thermal imaging, and a GPS enabling hand-carrying and UAV-mounting [56]. The device was used to monitor four tree properties, morphology, canopy volume, leaf area index, and fruit count, in apple and grape orchards. Peng et al. proposed an apple orchard mapping system [57]. They used a pair of coupled sensors of an RGB camera and LiDAR. The feature points extracted from the RGB image and point cloud were associated using a super-pixel matching technique. Zhang et al. proposed a branch segmentation algorithm for apple tree images [58]. Its aim was to measure the number of branches and extract the branch

topology and length. Point cloud data were acquired with backpack LiDAR (LiBackpack DG50). Based on the formal model of TreeQSM, the point cloud data were analyzed and segmented into a hierarchical tree topology of first, second, and third-order branches.

Wahabzada et al. proposed a tree segmentation method to extract various geometric shape properties from grapevine images [59]. A 3D laser scanning device was used to obtain the point cloud of a tree. A histogram was calculated, and k-means clustering was applied based on the histogram. Each cluster was classified into berry and rachis regions depending on geometric features. Scholer et al. proposed a method for reconstructing grape bunches in 3D and deriving useful phenotypic traits [60]. Using a laser scanner of Perceptron ScanWorks v5, they acquired the point cloud for a grape bunch consisting of berry, pedicel, and peduncle regions in the BBCH89 period, during which the berries were fully ripe for harvest. Since the stem was invisible in the season, scanning was performed twice, first scanning the complete grape cluster and then removing the berries and scanning the stem systems. The same research group extended the work to start the processing at an earlier stage, BBCH73, with groat-sized berries and mostly visible stems [61].

Zhu et al. proposed a tomato canopy reconstruction method using multi-views [62]. Using three Kinect v2 sensors placed 120° apart, three RGB-D images were acquired. The processing was performed in three different seasons, flowering, florescence, and fruiting, to monitor the change in phenotypic traits such as tree height, canopy width, and leafstalk angle. The same research group proposed an extended phenotyping system [63]. In this case, further segmentation of tree organs such as the stalk and leaf was performed, and finer traits related to the tree canopy were calculated.

Cheein et al. proposed a pear tree segmentation method for canopy volume estimation [64]. Four computational geometry methods, namely the convex hull technique, the segmented convex hull technique, cylinder-based modeling, and the 3D occupancy grid technique, were attempted. The system proposed in [65] had several sensors, namely LiDAR sensors, a stereo camera, an odometer, a GPS, and a tilt sensor. A peach tree row was scanned using LiDAR and a stereo camera mounted on a vehicle. The authors used a strategy to improve accuracy by fusing the obtained sensor data. To remove the ground points, the RANSAC algorithm was applied.

Underwood et al. proposed an approach for almond tree segmentation and 3D reconstruction of a tree row with the aim of documenting the traits of trees and later retrieving and updating them for tree management tasks [66]. The LiDAR data of a tree row was sliced and processed sequentially by a hidden semi-Markov model (HSMM). This model segmented the individual trees by transitioning among four states: tree, boundary, gap, and small tree. The same research group proposed an extended version, with the input being the top-view LiDAR point cloud covering an entire apple farm [67]. Based on the vehicle heading angle, the whole farm was divided into tree rows, and then each tree was processed using HSMM. This research group further improved the system by adding pixel density features of flowers and fruits [68]. Additionally, the same research group proposed a graph-based processing algorithm [69]. The point cloud data of avocado and mango trees were captured using a handheld GeoSLAM Zebedee 1 LiDAR sensor. Synthetic data generated from the SimTreeLS tool [219] was also utilized. Initially, the point cloud was converted into a voxelized data structure and then into a graph. A trunk node was identified, and the remaining nodes found their shortest paths to this trunk node. A method for branch segmentation was developed for phenotype extraction from apple trees [70]. The Laplace multi-scale adaptive algorithm was used to phenotype the plant height, crown width, branching number, and initial branching height. The dataset is publicly available and it is placed in Appendix A that has two tables, Tables A1 and A2.

3.3.2. Harvesting

Yongting et al. proposed an apple tree segmentation method for constructing a picking robot [71]. An image was captured with Kinect v2 and transformed into point cloud data using Kinect software. The feature sets of RGB, HIS, and fast point feature histogram (FPFH) from the point cloud were extracted. These features were classified into fruit, branch, and leaf categories using an SVM, whose hyperparameters were optimized using the genetic algorithm.

3.3.3. Spraying

Mahmud et al. proposed a tree segmentation algorithm to estimate the apple tree canopy foliage density and volume [72]. A sequence of point cloud data was acquired using a VLP-16 LiDAR sensor with a scanning speed of 5 frames per second. In preprocessing, points corresponding to the ground were removed using a RANSAC-based outlier detection algorithm. The tree area was subdivided into four vertical sections, and the number of points in each section was counted, excluding the trellis wire, pole, and trunk. Aiming to build a spraying machine, a method for measuring the canopy volume of apple trees was proposed [73]. The point cloud of a tree row was segmented into individual tree regions using the grid integration volume technique and partial least squares regression.

3.3.4. Pruning

Karkee utilized point cloud data obtained with a time-of-flight (TOF) camera (CamCube 3.0) to segment an apple tree image [74]. By leveraging the distance quality and amplitude information encoded in each pixel, background pixels were removed. The resultant 3D points were skeletonized using the medial axis algorithm. By examining the pruning rules of expert humans, a set of rules was developed to determine the pruning points on the skeletal tree structure. Elfiky et al. employed Kinect v2 to capture the front and back sides of an apple tree and generated point cloud data using Kinect Fusion software [75]. To merge two point clouds into one, the authors proposed an algorithm that combined skeletonization with the iterative closest point (ICP) algorithm. Zeng et al. proposed an apple tree segmentation method aimed at constructing a pruning or spraying machine [76]. An image was captured with a VLP-16 LiDAR sensor (Velodyne LiDAR, San Jose, CA, USA)from a distance of approximately 3 m, containing approximately 4500 to 8300 points per image frame. From this point cloud data, the background was removed using the RANSAC algorithm. Based on the fact that the trellis wires formed a plane, a simple rule was established and applied to detect the trellis.

You et al. proposed a cherry tree segmentation method for constructing a pruning robot operating during the dormant season [77]. This method was tailored specifically for upright fruiting offshoot (UFO)-shaped trees. The point cloud underwent skeletonization, and points were classified into five categories: trunk, support branch, leader branch, side branch, and none, utilizing both topological and geometric constraints. The same research group expanded upon this work by proposing improved decision-making for cutting points and conducting a field test [78]. Li et al. proposed a 3D reconstruction method for jujube dormant trees [79]. Three RGB cameras captured images that were utilized by the structure from motion (SfM) algorithm to reconstruct the dense point cloud. The SIFT descriptor was employed for feature matching.

Westling et al. proposed a segmentation method for mango and avocado trees for pruning tasks [80]. They applied a processing stage similar to that used in their previous study [69], which was aimed at handling phenotyping tasks. The previous study utilized a graph to identify the shortest paths from nodes to the trunk node and aggregated them to identify individual trees. A method for segmenting the trunks and branches from the point

cloud was proposed for apple and cherry trees [81]. The method used the adaptive cuboid region growing technique to segment 3D shapes incrementally. Aiming to build a pruning system for UFO-shaped cherry trees, a vision module producing and processing the point cloud data has been proposed [82]. The iterative closest point (ICP) algorithm was used to register and segment the point cloud into cordon, offshoot segment, and side branches.

3.3.5. Yield Estimation

Zine-El-Abidine et al. proposed an individual apple tree segmentation method for counting the apples on the tree [83]. They utilized two seasonal point clouds, one captured at harvest time and the other during the winter. Since the winter image revealed the entire branches, it facilitated the segmentation of individual trees in orchards. Using the harvest image, apples on the tree were detected based on thresholding color channels. They proposed a method that mapped the fruits onto the individual tree. Aiming to circumvent the use of expensive LiDAR sensors, Dey et al. employed the SfM technique to recover the color point cloud from an uncalibrated sequence of grapevine images [84]. The reconstruction was accomplished using the bundle adjustment algorithm and multi-view stereopsis. Salient features were extracted from the point cloud using both color and shape information. Using SVM, each point was classified into three categories: berry, branch, and leaf.

3.3.6. Navigation

Zhang et al. proposed a method for extracting tree row information and detecting trunks within the row to guide a navigational robot [85]. A sequence of tree row images was acquired using a custom-built LiDAR sensor. The point cloud was registered and refined based on odometry measurements using the iterative closest point (ICP) algorithm. The tree trunks were detected using the particle filter algorithm, which incorporated both the motion data from odometry and the point cloud information.

3.3.7. Thinning

Nielsen et al. proposed a method for reconstructing peach trees aimed at facilitating blossom thinning [86]. Three RGB cameras were utilized to capture the images. From these images, the point cloud of the tree was recovered using the trinocular stereo technique. The images were captured at night using high-intensity illumination, and the color information was mapped onto the point cloud to produce a color point cloud.

3.4. Others

3.4.1. Phenotyping

Chene et al. proposed a depth image segmentation algorithm for apple and yucca trees [87]. In a well-controlled indoor situation where each tree was isolated from other objects, they used Kinect to obliquely capture an RGB-D image of the tree. Only the depth map was used for segmentation. The maximally stable extremal region (MSER) algorithm was used to segment the depth map into individual leaf and branch regions.

3.4.2. Harvesting

With the aim of improving the success rate of robot picking, Jin et al. proposed a method to accurately localize the grape ears and cutting point of a grape stem [88]. They captured a far depth map using a binocular CCD camera and a close depth map using a RealSense camera installed above the robot hand. The color images from the binocular camera were analyzed and thresholded to identify the grape ear. The robot hand moved close to the grape stem and RealSense captured a close image.

Qiang et al. proposed a citrus tree segmentation algorithm to safely guide a picking robot in a complex natural environment [89]. A multi-spectral, five-channel image of a citrus tree was acquired using a custom-built camera. A reference spectrum was made by collecting ROIs corresponding to branch regions. A spectral angle mapper was used to convert the four channels into a single spectral angular value, α, by calculating the similarity between the image spectra and reference spectra. Branch regions were then identified by thresholding the α map.

Tanigaki et al. proposed a cherry tree segmentation method that used infrared and red laser sensors [90]. To make the problem easier to solve, they assumed "single trunk training," which results in a tall tree with fruits placed around the trunk. They used an infrared 830 nm laser beam to measure the distances to tree parts and a red 690 nm laser beam to detect the red fruit.

Bac et al. proposed a pepper plant segmentation method that used multi-spectral images for the obstacle avoidance of a picking robot [91]. An image was labeled with five classes: the stem, leaf top, leaf bottom, fruit, and petiole. The background was removed by thresholding the near-infrared wavelength. The same research group improved the system by enhancing the stem localization capability [92]. Colmenero-Martinez et al. proposed a trunk detection method for olive tree images to select the shaking point for a shake-and-catch harvesting machine [93]. An infrared LED scanner mounted on the end-effector was used to capture the image. A sequence of images was analyzed in real-time to decide where to guide the end-effector and when to stop and start shaking.

3.4.3. Pruning

Chattopadhyay et al. proposed an apple tree segmentation method that used multiple depth images acquired with a TOF sensor [94]. From an apple tree, multiple depth images were acquired at slightly different positions. Each depth image was thresholded to remove the background, and the results were cleaned using a noise removal technique. Using the clean depth image, the system extracted 2D skeletons and estimated the diameter and center of a cross-section of each skeletal pixel to obtain a 3D point cloud. Their dataset is publicly available and introduced in Table A2. The same research group presented another approach that used the same dataset [95]. The new approach used only one depth image to reconstruct the 3D tree model. The same research group presented another 3D tree modeling algorithm that used 40 depth images per tree acquired with an LMS 111 camera [96]. A split-and-merge clustering algorithm divided the points into three classes: trunk, junction, and branch points. The trunk and branch points were modeled with cylindrical shapes.

Botterill et al. developed a tree growing grammar set and used it to parse the grapevine image captured by a trinocular stereo camera [97]. To ease the edge detection of each RGB image, the vines were placed on a blue background. The cane edge segments, along with their thickness information, were detected using edge detection. Then, grammatical rules representing the tree structure were applied that recursively joined adjacent cane segments and joined the segment pairs into cane parts. The same research group extended their work into a complete system by including the processes of deciding the cut points and planning the obstacle-avoiding robot trajectory [98]. Luo et al. proposed a segmentation algorithm for the peduncles of grape clusters using depth images [99]. These depth images were captured by a binocular stereo camera. The peduncle was localized as a good candidate for a cutting point. The centers of grape berries were also identified. Then, the 3D spatial coordinates of the cutting points and berry centers were determined using the correspondence between the left and right images.

4. DL Algorithms for Tree Image Segmentation

This section has the same structure as Section 3. The only difference is that this section reviews the studies that used the DL approach. The studies reviewed in this section used the DL models explained in Appendix A.1.3, along with the model learning method.

Table 2 summarizes the 124 DL-based tree image segmentation papers. Sections 4.1–4.4 strictly follow the structure of Table 2. The point cloud method was the most commonly used method in studies based on the RB approach. However, in the DL approach, only a few studies used a point cloud. This may be because RGB or RGB-D images often yield adequate performance in many practical applications, though their effectiveness can vary depending on environmental complexity and task requirements.

A point cloud is much more expensive to acquire, and DL models using the point cloud method are not better than models using RGB images.

4.1. RGB

Because of the high accuracy of DL segmentation models based on RGB images, many researchers have used RGB images. Figure 5b and Table 2 demonstrate this fact. Some studies used RGB-D images as inputs but used a depth map for thresholding with the aim of removing the farthest trees. In this case, because core segmentation models use RGB images, this section introduces them. For example, the papers of [113,130,151] belong in this category.

4.1.1. Phenotyping

Gene-Mola et al. proposed a combination method that performed fruit segmentation using DL models and the 3D reconstruction of apple tree rows using SfM [100]. A mask R-CNN was employed for the fruit segmentation. For the 3D reconstruction of a tree row, several tens of images were acquired and used by SfM. Their image dataset is publicly available and shown in Table A2. Sun et al. proposed a trunk segmentation method for apple trees with the purpose of estimating the diameter of a grafted apple tree trunk [101]. An RGB image was labeled with the trunk region, where each trunk had a ground truth for the diameter measured 10 cm above the grafting position. SOLOv2 was employed as a segmentation model. Suo et al. proposed a seedling segmentation method with the aim of grading apple seedlings [102]. An RGB image was acquired by placing an apple seedling on white paper. The image was labeled with four classes for the root, rootstock, graft union, and scion. The BlendMask model was employed to segment the seedling image, while seedling quality grading was also performed. Zhao et al. proposed an apple tree segmentation method for estimating five phenotypes [103]. They used the heterogeneous cameras installed in a smartphone. An RGB image was labeled with boxes for the fruit, graft, trunk, and whole tree. Additionally, five phenotypes were manually measured and labeled. YOLOv5s was used to detect the boxes. Based on the detection results and binocular vision, five phenotypes were estimated. Dias et al. proposed a flower segmentation method for apple, peach, and pear tree images with the purpose of estimating the bloom intensity [15]. An RGB image was labeled with the flower region. An FCN was employed for the segmentation. A unique feature of this study was the pursuit of a versatile model by training it with apple tree images and then testing the same model with peach and pear tree images. The dataset for the apple tree images is publicly available and introduced in Table A2. The same research group proposed a flower segmentation method for apple, peach, and pear tree images for the purpose of yield prediction based on the flower densities [104]. Considering the scarcity of manually labeled data, they proposed a self-supervised learning (SSL) technique based on contrastive learning.

Xiong et al. proposed a citrus tree segmentation method [105]. An RGB image was labeled with three classes for the fruit, branches/leaves, and background. BiSeNetv1 was employed to segment the regions. The segmentation results supported a visual–inertial SLAM-based VINS-RGB-D module, which obtained a semantic point cloud and finally produced a 3D tree row map. Chen et al. proposed a method for segmenting apple tree organs such as buds, branches, and leaves to build a phenotyping system at the bud stage [106]. Aiming at cherry growth monitoring, a method for segmenting the plant time series images was proposed [107]. To reduce the labeling burden, the self-supervised contrastive learning algorithm was developed. A method for detecting target objects such as apple tree trunks, supporters, and persons was developed based on the YOLOv5 model [108]. Lightweight GhostNet and SIoU loss function were employed to improve the model accuracy. The YOLOv8 model was enhanced by incorporating modules such as dynamic snake convolution and multi-scale extended attention. Abdalla et al. proposed a genetic algorithm-based hyperparameter optimization of the CNN aiming at phenotyping apple trees [109]. The hyperparameters optimized are the momentum, regularization, learning rate, and three values related to network architecture. Gao et al. proposed a method for detecting the main stems of tomato trees by enhancing the YOLOR [110]. Multiple rotating bounding boxes and angular regression were incorporated to improve the accuracy. A comparative study was conducted to evaluate the segmentation models for measuring apple tree canopy volume [111]. The fully convolutional network (FCN) and short-term dense connection (STDC) were top-ranked. A method for segmenting individual apple trees in dense orchard rows was proposed [112]. A unique feature of this approach is that the segmentation algorithm is based on the prompt engineering technique.

4.1.2. Harvesting

Kang et al. used a Kinect sensor to acquire RGB-D images, but the apple picking robot they constructed only used RGB images [113]. They accomplished fruit detection using a modified gated feature pyramid network (GFPN). After the fruit detection, DasNet-v1 was used to segment the fruit and branch regions. The same research group extended their work by adopting DasNet-v2 with ResNet-101 as a backbone [114]. The model was trained to perform instance segmentation for fruit and semantic segmentation for stems. After the segmentation, the system constructed a 3D map that was used for determining the fruit pose and 3D working space of the robot. Wan et al. proposed a method for segmenting the branches in apple tree images with the aim of more accurate path planning for a picking robot [11]. To enhance the robustness of their dataset, the images were collected on sunny and cloudy days under various illumination conditions. Jiang et al. proposed a method for the segmentation and reconstruction of the thin wires located behind branches and leaves with the aim of avoiding these wires in the fruiting wall architecture of an apple orchard [115]. An RGB image was labeled with the wire regions. The BlendMask model was employed to segment the image. Wire pixels were stitched and fitted using a polynomial function to extract accurate skeletons. Kok et al. proposed an apple branch segmentation method with the aim of avoiding collisions with a picking robot [116]. An RGB image was labeled with branch and background regions. U-net++ was employed to segment the branch regions. Using an additional depth image and geometrical constraints, 3D branch surfaces were also reconstructed.

Kalampokas et al. proposed a grapevine segmentation method with the aim of building a grape picking robot [117]. An RGB image was labeled with three classes for the grapes, leaves, and background. Eleven different CNN models were trained, and their performances were compared. Their image dataset is publicly available and shown in Table A2. The same research group extended their study to include stem segmentation [118].

They showed that a regression CNN (RegCNN) model improved the accuracy. A trunk segmentation and diameter measurement method was developed to build a jujube harvesting robot [119]. The MobilenetV2, incorporating the convolutional block attention module, was employed and embedded in a mobile phone. Wang et al. proposed a grape fruit and nearby branch segmentation method [120]. An RGB image was labeled with three fruit varieties, including Muscat Hamburg, Chardonnay, and Summer Black. Additionally, it was labeled with three region classes for peduncles, branches, and leaves. The authors designed a new DualSeg network that was composed of a local-processing CNN and global-processing transformer modules. Wu et al. proposed a stem detection method for grapevine images [121]. A grape bunch was labeled with a box, and stem was labeled with three key points. YOLOv5 was used to detect the grape bunch and the HRNet was used to identify the stem key points.

With the aim of guiding a tomato picking robot without damaging other fruits or stems, Kim et al. proposed a 6D pose estimation method for tomato plants [122]. They combined an EfficientPose network and state-of-the-art 6D pose model into the DeepToMatoS network, which performed multiple tasks that included estimating the tomato maturity and identifying the 6D pose of the tomato and stem. Rong et al. proposed a method for tomato bunch detection and grasping pose estimation to improve the picking success rate [123]. YOLOv5m was employed to detect the tomato bunches and identify the maturity of an individual tomato. A picking sequence planning strategy based on the grasping pose information was also presented. Rong et al. proposed a tomato picking point detection method [124]. An RGB image was labeled with three classes for the fruit, calyx, and stem. Swin transformer v2 with an UperNet decoder was employed to segment the regions. With the aim of determining the easiest grasping poses for picking tomatoes, Kim et al. proposed a method to identify the poses of all the tomato and pedicel pairs in a scene [125]. An RGB image was labeled with four key points representing the fruit center, calyx, abscission, and branch. An OpenPose network was used as a backbone to detect the key points and estimate the multiple-tomato poses.

Liang et al. proposed a fruit bunch and stem segmentation method for litchi tree images to select picking points [126]. They adopted YOLOv3 to detect a fruit cluster area with a bounding box. U-net was used to segment and crop a 150 $\times$ 150 ROI centered in a bounding box. The system used night images with controlled illumination. Chen et al. proposed a 3D tree row reconstruction method for a variety of fruits [127]. Experiments were performed with litchi, guava, passion fruit, and citrus tree images. A unique feature of the method was the integration of stereo vision and SLAM techniques to reconstruct a 3D model of a tree row. Zhong et al. proposed a fruit and fruit-bearing branch segmentation method for the purpose of building a litchi picking robot [128]. An RGB image was labeled with fruit and fruit-bearing branch regions. YOLACT was trained using the labeled dataset. A module that determined the picking point and roll angle of a robot was also presented. Peng et al. proposed an accurate segmentation method for litchi branches with the aim of enabling clamping and shearing by a litchi picking robot [129]. To improve the segmentation accuracy, they developed a novel DL model called ResDense-focal-DeepLabv3+, which enhanced DeepLabv3+ by incorporating a ResNet and DenseNet with focal loss.

The guava picking system developed by Lin et al. used RealSense D435i, but only used the RGB channels [130]. The authors labeled 891 guava tree images with three classes for the fruit, branches, and background. They appropriately modified MobileNet to optimize it for the segmentation of these three classes and embedded it with channel and spatial attention modules. Wang et al. proposed a Sichuan pepper fruit detection and branch segmentation method [131]. An RGB image was labeled with a box for fruit and regions for the fruit and branches. A neural network inspired by YOLOP was developed to simultaneously

solve the multiple tasks of detection and segmentation. Zheng et al. proposed a trunk and branch segmentation method to identify an appropriate shaking point for a jujube tree [132]. An RGB image was labeled with three classes for the trunk, branches, and background. AGHRNet, which improved the HRNet by embedding a ghost attention module, was proposed to segment the regions.

Williams et al. proposed a kiwi picking robot [133]. The robot had four arms working simultaneously. It was an improvement of the system described in [220]. The robot worked in a kiwi orchard with a pergola style frame. The authors fine-tuned the pre-trained FCN to segment three classes for the calyx, cane, and wire. The calyx identification was advantageous in stereo matching and selecting gripping points. The same research group improved the robot system by using faster R-CNN and better end-effector [134]. Song et al. proposed a kiwi picking robot with a camera 1 m below the tree canopy looking upward [135]. Using DeepLabv3+ with a ResNet-101 backbone, the image was segmented into four classes for the calyx, branches, wires, and background.

Zheng et al. proposed a fruit segmentation and picking point detection method for mango harvesting [14]. An end-to-end DL model was proposed that simultaneously segmented the fruit region and detected the best picking point. A mask R-CNN was modified such that it had two additional outputs for the fruit region and picking point. Fu et al. proposed a bunch and stalk detection method for banana tree images [136]. An RGB image was labeled with a pair of boxes for the bunch and stalk. YOLOv4 was used to detect the boxes. The images were collected under various weather conditions such as sunny, cloudy, and overcast.

Wan et al. proposed a pomegranate tree segmentation method for a picking robot [137]. A total of 1000 artificial pomegranate tree images were synthesized, and 200 actual tree images were collected with RealSense D435. YOLOv4 with a CSPNDarknet53 backbone was employed to detect branches using bounding boxes. Li et al. proposed a fruit-bearing branch segmentation method for longan tree images for selecting the picking points [138]. An RGB-D image was acquired with RealSense D455 carried by a drone. The RGB image was input to the YOLOv5s model to detect the box that included the fruit and fruit-bearing branch. The box was further segmented into a branch region using DeepLabv3+.

Aiming to identify a grape picking point, an end-to-end model was proposed that simultaneously identified the grape bunch regions and keypoints [139]. The model employed the YOLOv8, incorporating the EMA (efficient multi-scale attention) module. An end-to-end model that simultaneously performs the tomato picking points detection, pose estimation, and obstacle segmentation was developed [140]. The YOLO model was improved to learn effectively those multi-tasks. To build a mango harvesting system, an improved YOLOv8 was proposed to simultaneously detect the fruits and fruiting stems [141]. A method for localizing the mango picking points was proposed [142]. YOLOv8 was improved by incorporating BiFPN and SPD-con module. The picking points were decided using a rule set. The YOLO and R-CNN models for fruit and branch segmentation were benchmarked using the mango datasets [143]. The additional modules for fruit size estimation and branch avoidance were also proposed. Sapkota et al. compared the performances of YOLO and R-CNN models for segmenting apple trees in dormant and early growing seasons [144]. A lightweight semantic segmentation model was proposed to segment the grape tree image into peduncles, branches, and leaves [145]. The rule set that identifies the picking points was also developed. A segmentation model for litchi fruits and branches named litchi-YOSO was proposed [146]. A rule set that localizes the picking points based on branch morphology reconstruction was also developed. To build a kiwi harvesting system in a trellis cultivation, a model for stem segmentation and picking point localization has been developed [147]. The U-Net model was improved by incorporating the

depth-wise separable convolution and spatial attention module. A segmentation model for accurately localizing picking points of grape bunches was developed [148]. The YOLO11 was improved by combining dynamic convolution and the Ghost module to localize the oriented bounding boxes accurately. Shen et al. proposed the multi-scale adaptive YOLO model performing grape pedicel instance segmentation [149]. Several ideas, such as a multi-scale attention head and a star fusion model, enhanced the segmentation accuracy. Wu et al. proposed the TinySeg model that segments grape pedicels for building a harvesting robot [150]. The model enhanced the accuracy by employing the SPD-conv module to expand the receptive field while preserving fine-grained details.

4.1.3. Spraying

Kim et al. developed a spraying system for a pear orchard [151]. The system captured RGB-D images with RealSense D435. They input the RGB images into SegNet and segmented them into five classes, including leaf–branch–trunk, fruit, ground, sky, and pipe regions. The same research group extended their work to have three spraying modes: all open, on/off, and a variable flow rate [152]. Their experiments showed that the three modes reduced the pesticide used to 56.8%, 39.37%, and 8.08% in areas where no tree was placed, respectively.

4.1.4. Pruning

Tong et al. proposed a trunk and branch segmentation method for the robotic pruning of apple trees [153]. A total of 2000 images were collected in the dormant season after the leaves had fallen off. An RGB image was labeled with three classes for the trunk, branches, and support. Two models that used a cascade mask R-CNN with ResNet-50 and Swin-T as a backbone were trained and evaluated. The same research group proposed an improved method that used SOLOv2 [154]. The paper presented a pruning point decision method that used a rule set that worked with the depth information obtained by a RealSense D435i camera.

With the aim of building a cane pruning system, Williams et al. proposed a grapevine segmentation method [155]. An RGB image was labeled with four classes for the trunk, cane, wire, and node. The image was segmented with a Detectron2 network. Using two segmented images of the left and right sides, a 3D tree model was constructed and used by a decision maker for cane pruning. Gentilhomme et al. proposed a grapevine branch segmentation method to assist in accurately understanding the vine structure and planning pruning points [156]. An RGB image was captured with a smartphone camera after placing a white curtain as the background. This RGB image was labeled with five classes, including those for the trunk, courson, cane, shoot, and lateral shoot. ViNet, which was an improved version of a stacked hourglass network (SHN), was proposed to identify the node positions and their relationships. Their dataset is publicly available and described in Table A2.

Borrenpohl and Karkee proposed a method to segment the trunk and leader regions in a cherry tree image [12]. In an orchard with an upright fruiting offshoot (UFO) tree architecture, two RGB tree images were captured with active and natural lighting during the dormant season. An RGB image was labeled with three classes for the trunk, leader, and background. A mask R-CNN was used to segment the regions. The YOLOv5 was improved to segment main stems, lateral branches, and fruit branches of tomato trees [157]. Using the segmentation result, a set of geometric rules localizes the pruning points automatically.

4.1.5. Yield Estimation

Hani et al. proposed an apple tracking method for apple counting in an apple tree row [158]. The work was an extension of [43]. While [43] used GMM, ref. [158] employed both GMM and U-net and compared their performances. They captured a video along a

tree row from the front and back sides. Each video was processed by U-net to segment the apples. By applying a tracking algorithm, the detected apples were tracked along a sequence of frames, and more accurate counting was accomplished. As a byproduct, a 3D reconstruction of the apple tree row was produced. Gao et al. proposed a fruit and trunk detection and tracking method for accurately counting apples [159]. An RGB image was labeled with boxes for the apples and trunk. The YOLOv4-tiny model was used to detect them. Unlike other detection-and-tracking approaches that tracked the fruits, this method tracked the trunk because the trunk was much larger and easier to track. The same research group improved their work by proposing a fruit tracking technique with mutual matching [160]. La et al. considered an orchard where adjacent trees were heavily intertwined and proposed a tree region segmentation method for apple tree images [8]. An RGB image was labeled with a tree region by drawing the contour of an individual tree. YOLOv8 was employed to segment this tree region. The dataset is publicly available and described in Table A2.

A berry detection and canopy segmentation approach was proposed with the aim of counting the actual number of berries in a grape cluster [161]. An RGB image was acquired and labeled with six classes: gap, leaf abaxial, leaf adaxial, shoot, trunk, and cluster. Using SegNet, a series of segmentation operations for the clusters, individual berries, and canopy elements were performed.

4.1.6. Navigation

Wen et al. proposed a segmentation method for pear and peach orchard images [162]. An RGB image was labeled with several classes, including road, person, fence, wall, ground, ladder, and fruit tree classes. Object segmentation outside the tree regions made robot navigation a potential application task. The authors proposed a transformer-based model called MsFF-SegFormer to segment the regions. To develop a vineyard row navigation robot, YOLOv8-based segmentation of tree trunks was proposed [163]. A path planning algorithm based on trunk positions was proposed.

4.1.7. Thinning

Hussain et al. proposed a green apple segmentation and pose estimation method with the aim of thinning the green apples [13]. An RGB image of green apples in the size of 10 to 30 mm was captured with a smartphone. The image was labeled with green fruit and stem regions. A mask R-CNN was used to segment the regions. Majeed et al. proposed a grapevine segmentation algorithm with the aim of selecting an end-effector pose to automate the green shoot thinning task [164]. Images were captured around the first week of the bud-opening season before the leaves appeared. Thus, the cordons were very visible. A total of 191 images were labeled using three classes for the trunk, cordon, and background. A pre-trained SegNet was transfer-learned to the tree image dataset. The same research group developed an extended system that dealt with grapevine images captured between the first week of bud opening and the fourth week when the cordon was only partially visible because of the foliage [165]. They used a faster R-CNN to detect the trunk and cordon bounding boxes.

With the aim of cutting off the male flower cluster of a banana tree, Wu et al. proposed a method for localizing the cut point [166]. An RGB image was labeled with rectangular boxes for the bananas, male flower clusters, rachis, and stem. YOLOv5s with an improved loss function was used to localize them. A cutting point was selected based on the depth information. With the aim of berry thinning of grape trees, the DeepLabv3+ model has been improved to segment the stem and berries at the fruit-sitting stage [167]. An algorithm for obtaining phenotypic characteristics of grapes enabled the identification of berries to be

removed. Aiming to build a thinning system, a method that segments the green apples was proposed [168]. An LSTM-based clustering algorithm was developed to pair the fruit stems at different times.

4.2. RGB-D

4.2.1. Phenotyping

Dong et al. proposed a 3D reconstruction method for an apple tree row and phenotyping method for measuring tree traits such as the tree volume, canopy volume, trunk diameter, and apple count [169]. Videos were captured with an Intel RealSense R200. Each of the front and back side RGB-D videos was processed to construct a 3D map, and the two resulting maps were merged into one. The SIFT features and RANSAC were used to process each video. A mask R-CNN was used to segment the trunk. Milella et al. proposed a grapevine segmentation method for canopy volume estimation and bunch counting [170]. Two RGB-D images were acquired using RealSense R200 devices mounted on a moving vehicle, one mounted laterally and the other in a forward-looking direction. These two RGB-D images were transformed into a point cloud. The point cloud was used to reconstruct a 3D tree row. Additionally, an RGB image was labeled with five classes: bunch, pole, trunk–cordon–cane, leaf, and background. VGG19 was employed to classify the patches into one of these five classes. A segmentation simulation for synthetic tree images was introduced in [171], whose results could be applied generally. The images for six tree species, including cherry and five non-fruiting trees, were synthesized using the SpeedTree software for tree modeling and the Unreal Engine for rendering. A total of 4800 images per species were generated. Each image was labeled automatically using six classes for the trunk, branches, twigs, leaves, ground, and sky. SegNet was used with three different inputs, including RGB, an early fusion of RGB and depth, and a later fusion of feature maps from the RGB and HHA networks.

A research team at Wageningen University and Research (WUR) published a series of papers proposing dataset synthesis for capsicum annum plants [172]. An image was labeled automatically with eight classes, including the background, leaf, pepper, peduncle, stem, shoot and leaf stem, wire, and cut peduncle regions. For comparison purposes, 50 actual images were captured and labeled manually. The same research group presented a segmentation algorithm for a synthetic dataset [173]. A practical training strategy was proposed in which a large synthetic dataset was used for bootstrapping a CNN model and a small actual dataset was used to fine-tune the model. They published another paper that improved the synthetic dataset using a generative DL model like cycle GAN [174]. A branch segmentation model for the apple trees in heavy occlusion was developed to build a phenotyping system [175]. The experiments showed that the proposed HOB-CNNv2 was superior to the conventional U-Net. To build a system that monitors growth status and overall health of tomato trees, Qi et al. developed a method that measures diameters of tomato main stems [176]. The YOLOv8 was improved by employing a soft-SPPF module to extract multi-scale features.

4.2.2. Harvesting

Zhang et al. proposed an apple tree segmentation algorithm with the aim of selecting the shaking points for a shake-and-catch system [177]. Images were acquired of a fruiting wall architecture using Kinect v2 during the dormant season. An image was labeled with two classes for the branches and background. The region maps from the pseudo-color image and depth image were combined to improve the accuracy. The regions were segmented using an R-CNN with AlexNet as the backbone. Because dormant season images are not appropriate for harvesting purposes and the input for AlexNet was small 32 $\times$ 32 images,

the same research group proposed an improved version for a shake-and-catch harvesting machine [178]. A total of 253 apple tree images were newly acquired using Kinect v2. An image was labeled with three classes for the trunk/branches, fruit, and leaves. A pre-trained Deeplabv3+ with ResNet-18 as a backbone was employed and fine-tuned. The same research group published another paper that extended the scope when selecting the shaking point [179]. In the new research, 785 new apple tree images were acquired using Kinect v2. An image was labeled with four classes for the branches, trunk, fruit, and background (mostly leaves). The same research group presented a different approach that used a faster R-CNN as the object (fruit, branch, and trunk) detection model [180]. The boxes for the branches and trunk were fitted with skeleton segments. The shaking point was then determined based on these segments. Granland et al. proposed a semi-supervised learning method for apple tree segmentation to accomplish multiple tasks, including harvesting, thinning, and pruning [181]. Because manual labeling is very expensive, a semi-supervised method that uses a small number of labeled samples and large number of unlabeled samples is needed. One way to accomplish this is to use human-in-the-loop labeling, where a model is initially trained with a small number of labeled samples, and then unlabeled samples are segmented, and a human corrects the results.

Coll-Ribes et al. proposed a bunch and peduncle detection method for grapevine images [182]. An RGB image was labeled with three classes for the bunches, peduncles, and background. The RGB and depth channels were concatenated and input to a mask R-CNN to segment the regions. The diameter of a bunch was measured and the cut point for a peduncle was selected based on the depth information. Xu et al. proposed a fruit and stem segmentation method for selecting the picking point for a cherry tomato plant [183]. An RGB image was labeled with fruit bunch pair, stem, and background regions. An improved mask R-CNN was employed. It used a combination of RGB and depth channels as the input and had two output heads for the fruit and stem. Zhang et al. proposed a 3D pose estimation method for a tomato bunch with the aim of guiding a picking robot [184]. The main stem, fruit stem, and fruit were modeled using a cylinder, two connected cylinders, and sphere, respectively. A box and 11 key points were labeled per tomato bunch to localize them. An Hourglass model was employed to detect them. The detected information was finally transformed into 3D pose information, which was used to guide the picking robot to cut the bunch. The same research group improved their work by proposing a detection method for invisible key points [185].

Li et al. proposed a litchi tree segmentation method for a picking robot [186]. A total of 452 images were collected with Kinect v2. An image was labeled with three classes for the twigs, fruit, and background. The twig regions were post-processed with morphological operations and skeletonization. After selecting the fruit-bearing twigs, 3D information was recovered using a point cloud. Yang et al. proposed a fruit detection and branch segmentation method for citrus tree images [187]. Images were collected under different lighting conditions, including front, side, and back lighting. Each image was labeled with the fruit and branch regions. A branch was labeled with two types of labeling, overall and segmental. A mask R-CNN model was used to segment the regions. The whole branch and trunk regions were identified by merging branch segments.

Lin et al. proposed a guava tree image segmentation and fruit pose estimation method for robot picking [188]. A total of 437 RGB-D images were collected by Kinect v2. An image was labeled with three classes for the fruit, branches, and background. First, an image was segmented into fruit and branch regions by fine-tuning an FCN with a VGG-16 backbone. By analyzing the geometric relationship between an individual fruit and mother branch, the fruit pose was estimated. The pose information was used for planning a collision-free robot path. The same research group improved the system by modifying a series of processing

stages [189]. They collected 304 new guava tree images. An image was labeled with three classes as in the previous paper. Because the dataset was small, an improved DL model, a tiny mask R-CNN that used a tiny CNN as a backbone with eight convolution layers and five pooling layers was employed. The fruit and branch regions were converted into a point cloud. The fruit regions were fitted with spherical shapes using RANSAC. In the successive paper, they also proposed a robot path planning algorithm that used the 3D fruit and branch information [190]. Because the future robot path was heavily dependent on the past path, the authors regarded the path planning problem as a Markov decision process. They adopted a reinforcement learning model and trained the model with a deep deterministic policy gradient (DDPG), coupled with long short-term memory (LSTM).

Yu proposed a pomegranate fruit detection method [191]. A mask R-CNN was used to segment an RGB-D image into fruit regions. The region information was converted into a point cloud, which was processed with PointNet. Finally, 3D boxes representing fruits and the surroundings such as branches and leaves were obtained. Ci et al. proposed a method that detects keypoints from RGB-D images and recovers the point cloud [192]. The point cloud was used to estimate the 3D pose of peduncle nodes and decide the picking point. Li et al. proposed a peduncle collision-free grasping posture for tomatoes [193]. They improved the HER-SAC (soft actor-critic with hindsight experience replay) reinforcement learning algorithm to generate action sequences for the harvesting robot dynamically. The dual-resolution network with a convolutional attention model for accurately localizing peduncle cutting points of tomatoes was developed [194].

4.2.3. Spraying

A citrus tree crown segmentation algorithm was proposed for the purpose of spraying [195]. A special aspect of this study was the use of images taken during different seasons that included the seeding, flourishing, and fruiting stages. A total of 766 images were acquired with Intel RealSense d435i. A mask R-CNN with an SE attention embedded ResNet as a backbone was developed. Each image was preprocessed to remove the regions at the back row using a thresholding depth map. To spray precise pollination for apple trees, Bhattarai et al. proposed an RGB-D image segmentation method [196]. The system identified target flower clusters and estimated their positions and orientations for planning the sprayer to apply charged pollen suspension to the target flower clusters.

4.2.4. Pruning

Chen et al. proposed an apple tree branch segmentation algorithm that used RGB-D images acquired with an Intel RealSense D435 [197]. The results could be applied to the branch pruning and fruit thinning tasks. They labeled 512 images with two classes for branches and non-branches. Four-channel RGB-D images were input into neural networks. Three models, Pix2Pix based on a generative adversarial network, U-net, and DeepLabv3, were compared and the superiority of U-net was shown. To identify trunks and branches of apple trees in the winter season, Ahmed et al. proposed a YOLOv8-based instance segmentation method [198]. The principal component analysis (PCA) was applied to measure the branch orientation and diameter.

4.2.5. Training

Trunk and branch segmentation is critical to accomplish automatic tree training. Majeed et al. proposed an apple tree segmentation algorithm with the aim of automating the tree training task [199]. The SegNet segmented the image into three classes for the trunk, branches, and trellis. They preprocessed an RGB-D image to obtain a foreground RGB image by removing pixels whose depth was greater than 1.3 m.

4.2.6. Navigation

Brown proposed a segmentation method of the RGB-D images of apple trees, aiming at building a navigating robot [200]. Using the particle filter algorithm, the method segmented the trunks and estimated the trunk sizes. For the autonomous navigation for tractors in peach orchards, a multi-task model that segments tree trunks, obstacles, and traversable areas was proposed [201]. The motion planning module used the depth map to decide the trajectory.

4.2.7. Thinning

An apple bud thinning method was developed to manage the crop load precisely based on bud detection and branch diameter measurement using YOLOv8 [202].

4.3. Point Cloud

4.3.1. Phenotyping

Dong et al. proposed a method for phenotyping the 3D traits of individual apples [203]. A UAV with three cameras obliquely captured images of an apple tree row. The images were captured using three different tree training systems. A dense point cloud was constructed using the multi-view structure obtained from a motion algorithm based on bundle adjustment. A modified U-net that used voxels as an input produced feature maps in the embedding space. A clustering algorithm was applied in the embedding feature space to segment individual apples. The apple count, volumes of individual apples, and spatial density distribution map were estimated as phenotypic traits.

Schunk et al. constructed a dataset to study the growth processes of tomatoes and maze and used it to develop a base DL model for tree segmentation [204]. They periodically captured point clouds for each growing stage and labeled them with two classes for leaves and stems. Tree segmentation was performed using PointNet. Then, spatio-temporal registration and surface reconstruction were attempted. The point cloud data was segmented to extract 3D phenotypes from apple trees [205]. The proposed FSDnet was based on density-based feature extraction and feature propagation. A method for estimating the length of the primary branch of apple trees in the deciduous period was proposed for phenotyping purposes [206]. The PointNet++ was employed to segment the point cloud into the primary branch, trunk, and endpoints of primary branches. A segmentation model for a point cloud of apple trees into a trunk and branches was proposed [207]. The PointNeXt model was improved by incorporating cylinder-based constraints.

4.3.2. Harvesting

With the aim of preventing a picking robot from colliding with other objects, Luo et al. proposed a method for grapevine segmentation and 6D pose estimation [208]. An RGB camera and LiDAR were used to capture grapevine images. From the RGB image, grape clusters were segmented using a mask R-CNN. A point cloud was then segmented based on the RGB segmentation results. Finally, the pose of a grape cluster was estimated by peduncle surface fitting, and the best cutting point was located on the peduncle ROI.

4.3.3. Pruning

Ma et al. proposed a jujube tree segmentation method for identifying a branch to be pruned, along with the best cutting point [209]. A jujube tree image was acquired with two Azure Kinect DK cameras mounted 35 cm apart and synchronized on a common supporting frame. First, the nearby trees and background were removed by thresholding the distance values and post-processing to obtain a clean point cloud. The reconstruction was accomplished by coarse registration based on skeleton points. The iterative closest

point (ICP) algorithm was applied for fine registration. The 3D tree was further segmented into individual branches to select the cutting points. An SPGNet with four steps that included geometric partitioning, super-point graph construction, super-point embedding, and contextual segmentation was adopted to achieve the trunk and branch segmentation.

To build an apple tree pruning system, the DeepLabV3++ model was used to segment RGB-D images [210]. The segmentation results from multi-view RGB-D images were registered to generate the point cloud to reconstruct the 3D tree branches. A multiple criteria decision-making method for pruning at various stages of cherry tree growing was proposed [211]. The PointNet++ was employed to segment the 3D branches to realize the method. Aiming to build a grapevine winter pruning system, a method for merging 2D segmentation information and point cloud data was proposed [212]. It extracts the thickness measurement and uses agronomic knowledge to place pruning points for balanced pruning. Using the point cloud of jujube trees, the PointNet++ model was trained and tested [213]. The integration of the Chebyshev graph convolution module improved the model. Aiming to build a pruning robot, a method for segmenting the point cloud of walnut trees was proposed [214]. The point cloud was reconstructed from multi-view images using a neural radiance field (NeRF). The PointNet-based model was used as a segmentation model.

4.4. Others

4.4.1. Phenotyping

Uryasheva et al. proposed a leaf segmentation method with the aim of monitoring the health of apple trees [215]. They used three cameras to obtain multi-spectral images of apple trees during the blooming season when the trees were most susceptible to fungal infection. An image was labeled with two classes for the leaves and background. An Eff-Unet, which combined an EfficientNet and U-net, was employed to segment the leaf regions of one-channel images. The segmentation maps from multiple channels were registered and used to calculate various vegetation indices (VI) such as NDVI, NDRE, and GNDVI. To support multiple tasks like picking, pruning, and pollinating, Liu et al. proposed a stem segmentation method for tomato plant images [216]. An RGB and NIR camera pair was used to capture images. The YOLACTFusion model, which integrated feature maps from the RGB and NIR images, was developed to enhance the feature discrimination ability. Experiments showed that fusing RGB and NIR images produced a superior segmentation accuracy.

4.4.2. Yield Estimation

Hung et al. proposed an almond tree segmentation method that used multi-spectral RGB–NIR images [217]. An image was labeled with five classes for the fruit, trunk, leaves, ground, and sky. Feature learning was performed in an unsupervised manner using a sparse autoencoder neural network. Instead of classifying each pixel, the correlation of neighbor pixels was considered, with a conditional random file (CRF) used for classifying pixels. A comparison of the RGB only and RGB–NIR images showed that the additional information provided by the NIR data enhanced the segmentation accuracy.

5. Discussion and Future Work

Based on the reviews in Sections 3 and 4, this section discusses the research trends and future perspectives regarding four aspects: sensors, methods, datasets, and generalizability. Finally, important future research directions are given.

5.1. Sensors

As shown in Figure 5b, the modern DL approach generally uses RGB or RGB-D images. This is primarily because a DL model guarantees a high segmentation accuracy when using RGB or RGB-D images. Actually, many studies captured RGB images using a smartphone. Because a farmer already has a smartphone, this would be very advantageous, with no extra cost for the sensor. For some tasks like picking, pruning, and thinning, an RGB-D sensor is essential because a robot needs 3D information for traveling to the target position. The most common sensor for RGB-D is an Intel RealSense D435, which costs approximately 200 USD. LiDAR sensors for point cloud data and sensors for multi-spectral or hyper-spectral images can be quite expensive, while the improvement in segmentation accuracy may be limited. Therefore, in the future, it is expected that more re searchers will use RGB or RGB-D images. One exception is when a very accurate phenotype is required because a LiDAR sensor provides more accurate 3D information. The theme of monocular depth estimation, which infers the depth from a single RGB image will be presented as a future work.

In the context of front-view orchard tree image segmentation, image resolution—both spatial and radiometric—plays a crucial role in determining the accuracy of segmentation algorithms. Higher-resolution images generally provide more detailed visual information, allowing for improved delineation of tree structures. However, they also require greater effort in terms of image capture, storage, transmission, and processing. This increased demand contributes to what is referred to as the digitization footprint, encompassing the environmental and computational costs associated with handling high-volume image data [221]. As a result, careful selection of image resolution and acquisition frequency is therefore essential to balance segmentation performance with computational and sustainability considerations, especially in large-scale or real-time orchard monitoring systems. While the current study is limited to ground-based front-view images, this trade-off becomes even more critical when the scope is expanded to include UAV or satellite imagery, where higher altitudes necessitate finer spatial and spectral resolutions to achieve comparable segmentation performance—further increasing the digitization footprint and the complexity of data processing workflows.

5.2. Methods

As explained in Figure 5a, the major solutions for tree segmentation are rapidly transforming from the RB to DL methodology. The DL model is changing from a CNN to transformer. Because a transformer has a less inductive bias than a CNN, the transformer is better for building a more versatile system. The fusion of a CNN and transformer will be presented in a future work.

The current tree segmentation methods are very specific to a given task and environment. Therefore, a new task in a new environment requires a high cost for designing, training, and testing a new method. This fact acts as a great barrier to broadly applying computer vision to tree segmentation. In the AI community, this kind of barrier is overcome by developing a uniform model with multi-modality capability. One notable example is Uni-perceiver, which can process text, image, and video data for classification, detection, and segmentation tasks using a single unified model [222]. In the agricultural domain, very limited attempts have been made in this direction. To the best of our knowledge, fruit or flower segmentation for several different species using a single model is the only example [104]. The development of a versatile model in the agriculture domain will be presented in a future work.

The 3D reconstruction of a row of trees is very useful for globally monitoring and managing orchard information. Combining it with individual tree segmentation will be

beneficial for many agricultural tasks. The reconstruction method still relies on RB such as SfM. The pursuit of DL-based reconstruction will be presented as a future work.

5.3. Datasets

A labeled dataset is an essential ingredient for building a successful DL model. In the medical imaging research communities, there have been many notable datasets, some acting as de facto standards [223]. In agriculture, there are surveys of datasets [5,224]. However, the list is deficient in quantity and quality. Because of the diversity resulting from numerous factors such as the fruit type, orchard environment, training type, time (day or night), season (dormant or foliage), image type, and viewing angle, the existing datasets listed in Table A2 of Appendix A.3 are limited in the sense that they only cover specific tasks and environments. The construction of versatile datasets will be presented as a future work.

Prominent recent themes in artificial intelligence are few-shot learning and self-supervised learning [225]. These learning strategies are very effective in overcoming the lack of human-labeled data because only a limited amount of labeled data is needed to enable a massive amount of labeling to be performed automatically. The employment of these learning methods for tree segmentation will be presented as a future work.

5.4. Generalizability

While many existing studies have demonstrated promising results, it is important to note that most experiments have been conducted on small-scale, narrowly scoped datasets, and field tests were often limited to specific orchard environments. As a result, the generalizability of current segmentation models remains a major concern. When these models are applied to different orchards with varying lighting conditions, tree architectures, and backgrounds, segmentation performance may degrade significantly. This highlights a critical limitation in current research and underscores the need for more robust solutions.

To address this, the importance of future work in two key areas is emphasized, as discussed in Section 5.4: (1) the construction of large-scale, diverse, and well-annotated benchmark datasets, and (2) the development of foundation models that can generalize across a wide range of orchard settings and fruit species. It is believed that progress in these directions will be essential to support the reliable deployment of segmentation systems in real-world agricultural environments, enabling more adaptive and accurate automation technologies in orchards around the world.

5.5. Future Works

Based on the three aspects mentioned above, six future works are summarized as follows.

- Table A2 in Appendix A.3 presents 11 public datasets related to tree image segmentation. Except for the last dataset, each one is highly specific to a given task. The construction of versatile datasets with large quantities and good quality is a critical element in future tree segmentation research. These datasets can serve as de facto standards to enable the objective comparison of newly developed segmentation models. Additionally, they will motivate challenging contests.
- If a foundation model is developed for agriculture, it could be applied to a wide range of tasks and conditions, either directly or with minimal fine-tuning [226]. Fine-tuning such a model would enhance its adaptability across diverse agricultural scenarios, including variations in season, soil type, and tree growth patterns. Specific constraints can be exploited when developing the foundation model for tree segmentation, including the fact that the objects in the image are confined to tree components such

as the fruit, flowers, branches, leaves, and trunk. These constraints will make the development much easier than that for a random scene.

- Fusing a CNN and transformer has been proved to extract more robust features [227]. Medical imaging is actively adopting fused models to identify thin vessels and small tissues [228]. Agricultural tasks could benefit from a fused model to improve the segmentation accuracy, especially in identifying small fruits, thin branches, and highly occluded fruits or flowers.
- Monocular depth estimation infers a depth map from a single RGB image or video. For natural images with applications in augmented reality or autonomous driving, many excellent methods are available [229]. However, few papers can be found in the agriculture domain. Cui et al. produced a vineyard depth map from an RGB video using U-net, which is a good basis to begin research in the agriculture domain [230]. The resulting maps could be used for tasks such as robot path planning and visual servoing.
- The 3D reconstruction of a row of trees could be implemented by combining DL techniques. For example, combining panoramic imaging and monocular depth estimation from an RGB video is expected to accurately generate a 3D tree row model.
- In the agriculture domain, this is just an initial stage of adopting few-shot learning or self-supervised learning [104,231,232]. These strategies will be very effective in overcoming the obstacles incurred as a result of the inherent large variations in agricultural tasks and environments. They will promote a more versatile system. It is highly recommended that a pre-trained model with a large and general agricultural dataset be built and then fine-tuned to handle downstream segmentation problems suitable for given tasks using few-shot learning.

6. Conclusions

This review conducted a comprehensive analysis of 207 research papers on fruit tree image segmentation, focusing on applications across various agricultural tasks. At the highest level of our taxonomy, studies were categorized into rule-based (RB, 83 papers) and deep learning (DL, 124 papers) approaches. This review further organized the literature according to image types, agricultural tasks, and fruit species, providing a structured overview of the field's evolution. Despite the technical progress reported in recent studies, many experiments have been performed using small-scale, narrowly scoped datasets, with field tests often limited to specific orchards. Consequently, current models may suffer from performance degradation when applied to other orchard environments with differing conditions such as lighting, tree architecture, and background complexity. This raises serious concerns about the scalability and robustness of existing approaches in real-world deployments.

To overcome these limitations, future research should prioritize two directions, as discussed in Section 5.4: (1) The development of large-scale, diverse, and well-annotated benchmark datasets that span multiple fruit species and orchard environments, (2) The construction of foundation models capable of generalizing across varied field conditions and agricultural tasks. We believe that these efforts will be instrumental in bridging the gap between laboratory results and real-world applications. Furthermore, due to the inherently domain-specific nature of fruit tree segmentation, interdisciplinary collaboration between computer vision researchers and horticulture experts is essential. Such collaboration will enable the design of more practical and effective segmentation systems and accelerate the deployment of high-accuracy automation solutions across diverse orchard settings.

Author Contributions: Conceptualization and original draft writing by I.-S.O.; validation and updating by J.-S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was supported by research funds of Jeonbuk National University in 2025.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

In the appendix section, three topics are described. Appendix A.1 explains the basic knowledge on image segmentation, including image type, segmentation methods, and performance metrics. Appendix A.2 presents various agricultural tasks supported by the tree segmentation. Appendix A.3 summarizes the agricultural datasets publicly available.

Appendix A.1. Basics of Image Segmentation and Performance Metrics

The localization of interesting objects in images is an essential problem for computer vision. This problem is solved in two different ways, detection and segmentation. A detection algorithm identifies an object using a rectangle, while segmentation designates a set of pixels covering an object. This paper focuses on segmentation.

This section briefly introduces the most popular image segmentation algorithms. These algorithms are divided into two methodologies, traditional RB and modern DL methods. In an RB method, humans manually design the detailed procedure of a segmentation algorithm using their own reasoning and heuristics. In contrast, a DL method trains a deep neural network using data. Because of its flexibility and high performance, DL became dominant in the early 2010s. Although RB methods now have diminishing value, they are included to present a complete review.

In general, an RB method is not *semantic* in the sense that it only outputs regions without object class information. In contrast, a DL method is semantic because it produces object class information with confidence values. Tree segmentation can be regarded as a special case of image segmentation because only one object class, "tree," is considered. Therefore, even the RB approach is semantic because an image is segmented into foreground and background regions, and the foreground is regarded as the tree object. Often, the foreground region is then further segmented into several parts such as the fruit, branches, trunk, and leaves.

In DL, general image segmentation aims at identifying numerous classes. For example, the YOLACT model can identify 80 classes of objects such as a person, car, bicycle, and cat. Two different types of segmentation are supported, semantic and instance segmentation. They are the same in the sense that object regions are segmented, and each region is labeled with an object class. However, when multiple instances of the same class are found in an image, semantic segmentation treats them as a whole by giving them the same identifier, while instance segmentation differentiates them by giving them different identifiers [233]. Depending on the requirements of an agricultural task, we may choose between semantic and instance segmentation. For example, in building a spraying robot aiming at reducing the amount of pesticide used by targeting only the tree regions, semantic segmentation is chosen. In building a pruning robot, instance segmentation is chosen because the target of the pruning is an individual tree.

Table A1 summarizes and compares the characteristics, advantages, and limitations of rule-based and deep learning methods.

Table A1. Comparison of rule-based and deep learning methodologies for tree image segmentation.

Aspect	Rule-Based Methods	Deep Learning Methods
Feature	Hand-crafted features (e.g., color thresholds, edge detection, texture analysis, and clustering) and predefined rules based on agricultural domain knowledge	Automatic hierarchical feature learning from agricultural data through deep neural networks such as CNN and transformer
Training	No training phase; rules defined by agricultural experts	Training a deep neural network using large labeled datasets
Robustness	Poor in varying lighting, occlusions, and complex backgrounds; struggles with overlapping trees, shadows, and clutter	Strong; handles variations in lighting, scale, occlusions, and complex spatial relationships
Generalization	Poor; specific to designed agricultural scenarios and conditions	Excellent; transfers well across different agricultural tasks with fine-tuning
Advantage	Fast, interpretable, no training data required, works well in controlled settings	High accuracy, robust to variations, handles complex patterns, scalable, semantic
Limitation	Sensitive to parameter tuning; fails in unforeseen conditions; not semantic	Requires substantial computational resources; needs large datasets

Appendix A.1.1. Image Type

Because of improvements in image sensing technologies, a variety of image sensors are available and various image types are used for tree segmentation. This paper classifies these into RGB, RGB-D, point cloud, and others. An RGB image consists of three channels with red, green, and blue wavelengths. An RGB-D image has an additional channel representing the depth. A point cloud image only stores the depths at points where the sensor can measure the distance of light travel. Usually, a point cloud is too expensive to acquire but provides more accurate depth values than RGB-D. Multi-spectral and hyper-spectral images belong to the "others" image type. A multi-spectral image requires more than three wavelengths, usually resulting in tens of channels. A hyper-spectral image has hundreds or thousands of channels. Sonar, thermal, and depth images also belong to this type.

RGB, RGB-D, multi-spectral, and hyper-spectral images are represented using a grid-like structure (i.e., a 3D array) because every pixel in the 2D image space has a value. In contrast, a point cloud image is represented by a 3-tuple set (x,y,d), where (x,y) is the image position and d is its depth at that position. Some sensors such as LiDAR cannot measure depths at some positions as a result of the physical limitations of laser light and use a point cloud.

Because an RGB camera is very cheap, with every farmer already possessing a high-resolution camera installed in their smartphone, and numerous high-performance deep learning models pre-trained with RGB images are publicly available, RGB images are the most popular. Since the first release of Kinect in 2010, many vendors have provided cheap and high-resolution RGB-D cameras at a cost of several hundred dollars. Kinect from Microsoft and RealSense from Intel are popularly adopted for agricultural applications. Because the LiDAR camera needed to produce a point cloud and a multi/hyper-spectral camera are very expensive, their usage is limited to special application environments. In the RB era, because segmentation algorithms were far from the performance level needed for practicality, some researchers directed their attention to point cloud or multi/hyper-spectral images, which provide richer information. However, in the DL era, this attention has dramatically diminished, and RGB and RGB-D images have become dominant. Currently, numerous deep learning models pre-trained with RGB or RGB-D images are available, and transfer learning to a tree image dataset is easily accomplished. Figure 5b and Table 2 show this trend. A specific survey paper focusing on RGB-D for fruit localization is available [234].

Appendix A.1.2. Rule-Based Methods

From the beginning of the computer vision era, researchers have developed many image segmentation algorithms [235]. The basic strategy is to group nearby pixels with similar intensity, color, and/or texture features into a region or to divide the image into successive regions by delineating contours where the features abruptly change. Some of the methods that have been popularly used for tree segmentation are briefly explained below.

Thresholding: The thresholding algorithm divides the pixels into two groups below and above a threshold. One of the two groups is taken as the foreground (tree region), and the other is considered to be the background. In implementing the thresholding algorithm, the most important decision concerns the optimal threshold value. Intuitively, a valley point in a histogram is a good candidate, but noise prevents this scheme from working successfully. Otsu proposed an optimization method that minimizes the weighted sum of the variances of the pixel groups in the foreground and background regions. Because the thresholding operation is applied to a single map, an appropriate channel for the RGB or RGB-D images must be selected.

Clustering: A pixel or super-pixel is represented by t-tuples (v_1,v_2,. . .,v_t), where v_i is a color or position value. The clustering algorithm groups the pixels or super-pixels into several clusters. Each cluster represents foreground or background region. Often a finer segmentation is accomplished by assigning each cluster to one of the branch, trunk, leaf, fruit, or background classes. Many clustering algorithms are available, such as k-means, Gaussian mixture model (GMM), fuzzy, and mean-shift.

Region growing: Starting from a seed region, the algorithm merges neighboring pixels under some constraints to grow the region. The snake algorithm is regarded as belonging to the region growing category because it successively expands the object contour by optimizing the smoothness and objectness of regions.

Machine learning: After representing a pixel with a t-tuple, as in the clustering approach, a machine learning model is trained with a human-labeled training set. At the inference stage, the pixels in a new image are classified into several classes such as branch, fruit, leaf, and trunk classes. The Bayesian, support vector machine (SVM), multi-layer perceptron (MLP), and random forest are popularly used learning models. The main difference compared to DL is that the model here has a shallow architecture (i.e., 1–3 layers), while DL models have tens or hundreds of layers. Because the layers are shallow, the feature learning is weak. Therefore, the feature vector is designed by humans heuristically. In contrast, DL models extract the optimal features by learning, resulting in a feature learning capability.

Fitting: Assuming geometric shapes such as lines, circles, and cylinders for the tree parts, a fitting method finds the optimal parameters for those shapes for the given point set. The Hough transform, random sample consensus (RANSAC), and geometric fitting algorithms belong to this approach.

Graph-based: A pixel becomes a node of a graph, and adjacent nodes are connected by edges whose weights represent the similarity between two pixels. A graph cut algorithm is applied recursively to the graph to partition the nodes into groups. Each group is regarded as a region. Sometimes, a line segment obtained using a skeletonization algorithm becomes a node, and node merging is applied successively to obtain a tree structure.

Appendix A.1.3. Deep Learning Methods

Figure A1 illustrates the basic input and output relationship of a DL model used for fruit tree image segmentation. In this case, the output map is represented using three channels for the branches, fruit, and background. Some other region maps are illustrated in Figure 2.

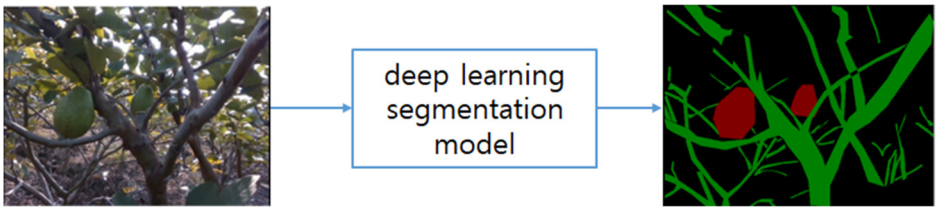


Figure A1. Outline of DL segmentation (a guava tree image is segmented into three classes: fruit, branches, and background [130]).

In 2012, AlexNet achieved a top-5 classification error rate (15.3%) in the ImageNet large scale visual recognition challenge (ILSVRC) and won first place [236]. This event motivated most computer vision research groups to change their methodology from RB to DL. After this successful application of a CNN to image classification, the first successful object detection model (R-CNN) [237] was proposed in 2014, and its performance continues to improve [238]. The R-CNN is a two-stage model that first generates numerous region proposals and then selects the true regions from them. Because of its generate-and-select strategy, it is slow. In 2017, the R-CNN was transformed into a segmentation model called a mask R-CNN, which has two heads, one for detection and the other for segmentation [239]. Since its advent, the mask R-CNN was popularly used for segmentation purposes. In 2016, a one-stage detection model called YOLO was proposed [240]. It is fast because it uses one stage of processing. The performance of YOLO has continued to improve from v1 to v8. Like the mask R-CNN, YOLOv8 is a dual-function model for detection and segmentation. In contrast to the R-CNN and YOLO models, which started as detection models and evolved into segmentation models, a fully convolutional network (FCN) was born in 2015 as a segmentation model [241]. The FCN was designed to use an original image as an input and output a segmentation map. After the success of the FCN model, many variants such as DeConvNet, U-net, and DeepLabv3+ have been proposed and popularly used. After the introduction of a transformer model for language processing by [242], the transformer model has successfully evolved to achieve image segmentation.

CNN-based models follow.

mask R-CNN: The R-CNN was developed as an object detection model and has evolved into fast R-CNN and faster R-CNN [243]. The pipeline of these models consists of a backbone, region proposal, and head. The backbone module extracts a rich feature map. The region proposal module generates several candidate patches with a high probability to be meaningful objects. The head module produces the "class" (object class and confidence information) and "box" (object location) branches. Adding the "mask" branch results in a mask R-CNN, which can simultaneously detect and segment the objects [239].

YOLOv8: In contrast to a slow two-stage R-CNN that generates many region proposals and successively evaluates each one, YOLO uses a one-stage process that directly regresses the object locations from each window in a grid-partitioned image [244]. Owing to its simpler architecture, YOLO achieves real-time processing while sacrificing accuracy. The original YOLOv1 has been continually improved by the originator and other research groups. The latest version, YOLOv8, expands the detection-specific model into a versatile model that enables detection, segmentation, pose estimation, and tracking [245].

YOLACT: YOLACT is the same as YOLO in that it uses one-stage processing and guarantees real-time processing. However, it uses a completely different approach with two modules, one generating prototypes and the other computing the mask coefficients [246]. The modules work in parallel, resulting in a high speed. The prototypes are feature maps of the same size as the input image, each of which is likely to represent the appearance of certain objects. Linearly combining the prototypes weighted by the mask coefficients results in the final segmentation map.

FCN: The input to an FCN is the original image and the output is a *k*-channel binary map, where each channel represents the regions of an object class. In the three-class segmentation shown in Figure A1, a three-channel map is used for the fruit, branch, and background regions. The encoder part of an FCN reduces the resolution, and the decoder part restores the original resolution. This restoration is achieved using the transpose convolution.

U-net: A U-net is a variant of an FCN, which was originally developed for medical image segmentation [247]. It adds a skip connection in order to enrich the feature maps transferred from the encoder to decoder. At present, the U-net is used universally for images generated in many fields, including tree images.

The transformer model is innovative in extracting rich feature maps by measuring the self-attention between words in an input sentence [242]. It has revolutionized natural language processing. The vision community modified the transformer model to make it suitable for processing images by considering the grid patches to be words and succeeded in obtaining superior performances compared to a CNN in many vision problems like image classification, object detection, segmentation, and tracking [248]. Well-known transformer models with segmentation capability are introduced here.

DETR: The detection with transformer (DETR) model extracts a *d*-channel feature map with ResNet [249]. Each channel is regarded as a word and input into the encoder block, which computes a self-attention map using a query–key–value operation. After passing several encoder blocks, the feature map is passed to a series of decoder blocks, where the feature map is transformed into object location and class information.

Swin transformer: The Swin transformer was developed to serve as a backbone for various tasks such as classification, detection, and segmentation [250]. It employs hierarchical multiscale feature maps, which start from small patches and gradually merge with neighboring patches in deeper layers. The algorithm employs the shifted-window concept, allowing the overlapping of window partitions.

Model training and performance metrics are explained in the following.

Model training: A tree segmentation model segments a tree image into 2–6 region classes. For example, La et al. segmented an apple image into two classes, the foreground (tree) and background, as seen in Figure 2a [8]. Hussain et al. segmented an apple image into three region classes, the fruit, stem, and background, as seen in Figure 2d [13]. It is a common practice for researchers to select an appropriate DL model and modify its head to suit the number of region classes.

In the training phase, two approaches are available: transfer learning and learning from scratch. Transfer learning fine-tunes a pre-trained deep learning model using a tree segmentation dataset. It keeps the weights of the feature extraction layers of the pre-trained model while initializing the weights of the new head layer. The fine-tuning does not significantly perturb the weights of the existing layers by keeping the learning rate very low, while setting the learning rate for the newly attached head layers to be relatively large. The low learning rate conserves the feature extraction capability of the existing layers. The learning-from-scratch method initializes the weights of all the layers and starts learning from scratch. A reasonable learning rate is applied over all of the layers.

Although there are several public datasets, as explained in Appendix A.3, most studies use private datasets. This is because the existing datasets are specific to a given task and particular situation. The construction of a general dataset seems to be the most important future work, and this issue is discussed in Section 5. Because the private datasets are small, data augmentation is very important. Usually, a combination of various geometric transformations such as rotation or flipping and various photometric transformations such as adding noise or an intensity change is applied to augment the data.

Performance metrics: Once the model has been trained, a performance evaluation follows. Several metrics are popularly used. The pixel accuracy (PA) is measured based on the ratio of correctly labeled pixels to the total number of pixels. The mean pixel accuracy (MPA) is defined as the average PA over all the classes. The intersection over union (IoU) is defined as the area of intersection between the predicted region and ground truth region divided by the union of the two regions. When a threshold value for the IoU is fixed, each region can be regarded as true positive (TP), false positive (FP), or false negative (FN). The precision, recall, and F1 score can be computed based on the numbers of TP, FP, and FN regions. Additionally, the average precision (AP) and mean average precision (mAP) are computed by varying the threshold. It is common for AP@0.5 and AP@0.75 to be provided for thresholds of 0.5 and 0.75, respectively. The AP@0.5:0.95 is also used, which represents the average AP using thresholds from 0.5 to 0.95 in increments of 0.05.

Appendix A.2. Agricultural Tasks Supported by Tree Segmentation

It is important to understand the agricultural tasks that require tree segmentation. Table 1 lists these tasks. Appendix A.2.1 explains the agricultural environments in which tree images are acquired. Appendix A.2.2 briefly describes each of the tasks in relation to tree segmentation.

Appendix A.2.1. Agricultural Environments

There are several environmental factors that influence the design and implementation of a tree segmentation algorithm.

Season: It is important to select the most appropriate season based on the task. Farmers perform pruning during the dormant season. Therefore, for the pruning task, images are typically acquired during the dormant season when the trees have no leaves. This is advantageous for segmenting the branches and selecting the cutting points. Figure 2c shows an example. For the spraying task, it is necessary to process an image taken at full foliage. Figure 2a shows an example.

Natural vs. trained tree: A modernized orchard trains the trees to have a well-controlled shape. This training is advantageous for the health and growth of fruit because it allows for good air flow and sunlight. Additionally, the simpler and rather flat canopies for the trained trees facilitate automating many tasks such as harvesting, pruning, and thinning. A naturally growing tree has a spherical shape, where the back side and interior of the sphere are typically hidden. This makes the segmentation of the fruit, branches, and leaves difficult. The delineation of neighboring trees is also very difficult when the trees are intertwined.

Day vs. night: In the daytime, the illumination varies greatly as the weather or viewpoint changes. When a robot changes its pose, the camera can face the sun or becomes backlit. The shadows also matter. These illumination factors degrade the performance. To cope with this difficulty, some researchers use night vision with artificial illumination. For example, Xiong et al. proposed a vision algorithm for a nocturnal image with LED illumination [27].

Growing stage: A fruit tree passes through a long period from the bud to the mature fruit. In the early stages, thinning is a major task, which removes some of the buds, flowers, or fruit. For example, Hussain et al. proposed a fruit and stem segmentation algorithm for thinning the green fruit of an apple tree, as shown in Figure 2d [13]. For the harvesting task, images taken at the mature fruit stage are used.

Appendix A.2.2. Agricultural Tasks

Next, the characteristics of each agricultural task are briefly introduced, along with several recent survey papers. These survey papers are valuable in overcoming the limit of our review and deepening the reader's insight and knowledge.

Phenotyping: Tree phenotyping is the task of describing the expression of tree traits [251]. In their review, Huang et al. grouped various traits into five aspects: water stress, architecture parameters, pigments and nutrients, degree of disease, and biochemical parameters [252]. The accurate segmentation of individual trees is sufficient to estimate some traits such as the tree height or canopy volume. For other traits such as leaf pigment, disease, or the branching shape, finer segmentation of the fruit, leaves, and branches is required. The academic and industrial activities of the International Plant Phenotyping Network (IPPN) are noteworthy, including the Forest Phenotyping Workshop.

Harvesting: Fruit harvesting can be divided into two approaches: bulk harvesting using the shake-and-catch method and selective harvesting by robot picking. The most recent survey papers discuss these [218,253–255]. In robot picking, the most important task is fruit detection. Finer segmentation is required to successfully access the fruit without colliding with other fruit, branches, or poles. In the shake-and-catch method, the segmentation of the trunk or main branch is sufficient.

Spraying: The main purpose of autonomous spraying machines is to reduce the amount of pesticide by precisely spraying the target tree. The nozzle of the spray gun is only open when it is moving across the tree region [256,257]. Because spraying is usually performed by navigating a tree row, real-time tree segmentation is required.

Pruning: According to [258], the harvesting and pruning performed during apple production account for 59% and 20% of the labor cost, respectively. Therefore, the automation of the pruning task is very important. In this task, the precise identification of branches is very important to analyze the branching structure and select cutting points [258–260]. The branch structure is analyzed, and the heuristics that farmers use are applied to the branch structure to select the cutting points.

Yield estimation: Direct yield estimation is the task of estimating the total amount of fruit on a tree or in a tree row by counting the visible fruits [261–263]. Tree segmentation is essential to perform fruit counting on individual trees. A video is used for fruit counting in a row of trees [264]. The tree regions should be precisely segmented in order to exclude the fallen fruit and fruit in other rows.

Navigation: Tree segmentation is required to plan a precise path for vehicle navigation along a tree row in an orchard [265,266]. For tall trees like palms, trunk segmentation is performed. For small trees like apple trees, whole tree regions should be segmented. It is common for information from multiple sensors such as GPS and odometry sensors to be combined with visual information to optimize the path planning.

Thinning: Thinning refers to removing some of the blossoms, flowers, or fruit to maximize the quality of the harvested fruit [267]. A traditional non-selective method such as a rotating string thinner carries the risk of damaging the tree. Often the thinning results are unsatisfactory because the same operation is enforced over the whole tree without evaluating the quality of each fruit or flower. A vision-based thinning method is selective because it evaluates each one and removes the bad ones. This method should segment each blossom, flower, or fruit and evaluate them.

Training: In modern orchards, it is common for fruit trees to be trained to have a controlled shape, which results in a high yield and good fruit quality. The automation of tree training requires the segmentation of branches, leaves, trellises, and poles. For example, Majeed et al. proposed a segmentation method for trunks, branches, and trellis wires to automate apple tree training [199].

Appendix A.3. Public Datasets

This section introduces the public datasets related to tree image segmentation that are readily downloadable from the web. All the datasets except the last row were constructed and used by the authors of the papers reviewed in Section 4. Table A2 summarizes them. Four and three of the 11 datasets are for apple trees and grape vines, respectively. One dataset each was found for tomato plants, avocado trees, and capsicum annum plants. All of these except the capsicum annum dataset contain real images. The last dataset, Urban Street Tree, was not constructed in an orchard, but on the street. However, it is included because fruit trees are available, and the quantity and quality of the dataset are good.

Appendix A.3.1. Apple

The WACL dataset constructed by Purdue University contains depth images captured from one indoor and four orchard trees [94]. Diameter information is also provided for each branch in a tree image. The diameter was used to determine whether the branch should be pruned and to select the cutting point. Fuji-SfM is an RGB dataset for apple trees constructed by Gene-Mola et al. at the University of Lleida [268]. The purpose of the dataset was to reconstruct a 3D tree row from several tens of RGB images. The same research group proposed a reconstruction algorithm and evaluated the performance using the dataset [100]. The NIHHS-JBNU dataset was constructed through the collaboration of the National Institute of Horticultural and Herbal Science and Jeonbuk National University, Korea [8]. Each RGB image was captured with the target tree in the center of the image. The images were captured in an apple orchard where adjacent trees were heavily intertwined. It was labeled with two regions for the trees and background. The dataset is known to be the first that considered intertwined trees. The Fruit Flower dataset was constructed at Marquette University by Dias et al. [15]. Each RGB image of an apple, peach, or pear tree is labeled with the flower region. To assist in this labeling, a Monte Carlo region growing technique was developed and used [269]. It is worth noting that the research group applied innovative ideas for a versatility test and self-supervised learning using their datasets [15,104].

Appendix A.3.2. Grape

The Grapes-and-Leaves dataset was constructed at the International Hellenic University by Kalampokas et al. [117]. Each RGB image is labeled with three classes for the grapes, leaves, and background. The same research group also produced the Stem dataset, which labels the stems of grape clusters [118]. The purpose of these two datasets was to build a grape picking robot by developing a deep learning model that selected the picking point and planned a safe path for robot travel. The 3D2cut Single Guyot dataset was constructed at the Idiap Research Institute, Switzerland, by Gentilhomme et al. [156]. It has 1513 images that are labeled with five classes for the trunk, courson, cane, shoot, and lateral shoot. Additionally, their node, termination, and spatial dependency are labeled. The purpose of the dataset was to build a pruning machine for a grapevine trained in the guyot system.

Appendix A.3.3. Tomato

The Pheno4D dataset was constructed with the aim of studying the growth processes of tomatoes and maze at the University of Bonn by Schunck et al. [204]. Seven tomato plants were recorded over three weeks, resulting in 140 point clouds. The images are labeled with two classes for leaves and stems. By separately labeling each leaf, instance segmentation for a leaf was possible. Because the images were captured indoors with the plants growing in separate pots, their application is limited.

Appendix A.3.4. Avocado

The Avocado dataset, which has a point cloud format, was constructed at the University of Sydney by Westling et al. [270]. Three avocado trees were captured in an image and labeled with three classes for the leaves, branches, and ground. The images at three different stages are provided, one before any pruning, a second after limb removal, and a third after limb removal and hedging. The dataset was used to build a phenotyping method [69] and an automatic pruning machine [80]. The research group also constructed a mango dataset, but they only released the avocado dataset.

Appendix A.3.5. Capsicum Annum (Pepper)

The Capsicum Annum Image dataset was constructed at Wageningen University and Research (WUR) by Barth et al. [172]. A dataset of 10,500 RGB-D images was synthesized from 42 plant models by varying plant parameters. An image is labeled with eight classes, including the background, leaf, pepper, peduncle, stem, shoot and leaf stem, wire, and cut peduncle classes. Because the renderer knew the class of each pixel, the labeling of the synthetic images was performed automatically. For comparison purposes, 50 actual images were captured and labeled manually. It was reported that approximately 30 min was needed to label one image. The dataset was used by the same research group for phenotyping [173]. The synthetic dataset was improved using a cycle GAN [174].

Appendix A.3.6. Street Trees

The Urban Street Tree dataset includes 41,467 images of 50 tree species, including some fruit-bearing trees. The 22,872 images are divided into several groups, and each group is labeled with a combination of trunk, leaf, tree, branch, flower, and fruit. The authors argued that the dataset could be used for benchmarking multiple agricultural tasks.

Table A2. Public datasets for tree image segmentation.

Dataset	Fruit	Image	URL
WACL [94]	apple	depth	https://engineering.purdue.edu/RVL/WACV_Dataset (accessed on 23 October 2025)
Fuji-SfM [268]	apple	RGB	http://www.grap.udl.cat/en/publications/datasets.html (accessed on 23 October 2025)
NIHHS-JBNU [8]	apple	RGB	http://data.mendeley.com/datasets/t7jk2mspcy/1 (accessed on 23 October 2025)
Fruit Flower [15]	apple, peach, pear	RGB	https://doi.org/10.15482/USDA.ADC/1423466 (accessed on 23 October 2025)
Grapes-and-Leaves [117]	grape	RGB	https://github.com/humain-lab/Grapes-and-Leaves-dataset (accessed on 23 October 2025)
Stem [118]	grape	RGB	https://github.com/humain-lab/stem-dataset (accessed on 23 October 2025)
3D2cut Single Guyot [156]	grape	RGB	https://www.idiap.ch/en/scientific-research/data/3d2cut (accessed on 23 October 2025)
Pheno4D [204]	tomato and maize	point cloud	https://www.ipb.uni-bonn.de/data/pheno4d/ (accessed on 23 October 2025)
Avocado [270]	avocado	point cloud	https://data.mendeley.com/datasets/d6k5v2rmyx/1 (accessed on 23 October 2025)
Capsicum Annum Image (synthetic) [172]	Capsicum annum	RGB-D	https://data.4tu.nl/articles/_/12706703/1 (accessed on 23 October 2025)
Urban Street Tree [271]	Street trees (50 species)	RGB	https://ytt917251944.github.io/dataset_jekyll/ (accessed on 23 October 2025)

References

1. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep Learning in Agriculture: A Survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
2. Thakur, A.; Venu, S.; Gurusamy, M. An Extensive Review on Agricultural Robots with a Focus on Their Perception Systems. *Comput. Electron. Agric.* **2023**, *212*, 108146. [CrossRef]
3. Hua, W.; Zhang, Z.; Zhang, W.; Liu, X.; Hu, C.; He, Y.; Mhamed, M.; Li, X.; Dong, H.; Saha, C.K.; et al. Key Technologies in Apple Harvesting Robot for Standardized Orchards: A Comprehensive Review of Innovations, Challenges, and Future Directions. *Comput. Electron. Agric.* **2025**, *235*, 110343. [CrossRef]
4. Dhanya, V.G.; Subeesh, A.; Kushwaha, N.L.; Vishwakarma, D.K.; Kumar, T.N.; Ritika, G.; Singh, A.N. Deep Learning-Based Computer Vision Approaches for Smart Agricultural Applications. *Artif. Intell. Agric.* **2022**, *6*, 211–229. [CrossRef]
5. Luo, J.; Li, B.; Leung, C. A Survey of Computer Vision Technologies in Urban and Controlled-Environment Agriculture. *ACM Comput. Surv.* **2023**, *56*, 118. [CrossRef]
6. Zhang, A.; Lipton, Z.C.; Li, M.; Smola, A.J. Dive into Deep Learning. 2023. Available online: https://d2l.ai/ (accessed on 12 October 2025).
7. Chehreh, B.; Moutinho, A.; Viegas, C. Latest Trends on Tree Classification and Segmentation Using UAV Data—A Review of Agroforestry Applications. *Remote Sens.* **2023**, *15*, 2263. [CrossRef]
8. La, Y.-J.; Seo, D.; Kang, J.; Kim, M.; Yoo, T.-W.; Oh, I.-S. Deep Learning-Based Segmentation of Intertwined Fruit Trees for Agricultural Tasks. *Agriculture* **2023**, *13*, 2097. [CrossRef]
9. Cheng, Z.; Qi, L.; Cheng, Y.; Wu, Y.; Zhang, H. Interlacing Orchard Canopy Separation and Assessment Using UAV Images. *Remote Sens.* **2020**, *12*, 767. [CrossRef]
10. Xiao, F.; Wang, H.; Xu, Y.; Zhang, R. Fruit Detection and Recognition Based on Deep Learning for Automatic Harvesting: An Overview and Review. *Agronomy* **2023**, *13*, 1625. [CrossRef]
11. Wan, H.; Zeng, X.; Fan, Z.; Zhang, S.; Kang, M. U2ESPNet—A Lightweight and High-Accuracy Convolutional Neural Network for Real-Time Semantic Segmentation of Visible Branches. *Comput. Electron. Agric.* **2023**, *204*, 107542. [CrossRef]
12. Borrenpohl, D.; Karkee, M. Automated Pruning Decisions in Dormant Sweet Cherry Canopies Using Instance Segmentation. *Comput. Electron. Agric.* **2023**, *207*, 107716. [CrossRef]
13. Hussain, M.; He, L.; Schupp, J.; Lyons, D.; Heinemann, P. Green Fruit Segmentation and Orientation Estimation for Robotic Green Fruit Thinning of Apples. *Comput. Electron. Agric.* **2023**, *207*, 107734. [CrossRef]
14. Zheng, C.; Chen, P.; Pang, J.; Yang, X.; Chen, C.; Tu, S.; Xue, Y. A Mango Picking Vision Algorithm on Instance Segmentation and Key Point Detection from RGB Images in an Open Orchard. *Biosyst. Eng.* **2021**, *206*, 32–54. [CrossRef]
15. Dias, P.A.; Tabb, A.; Medeiros, H. Multispecies Fruit Flower Detection Using a Refined Semantic Segmentation Network. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3003–3010. [CrossRef]
16. Snyder, H. Literature Review as a Research Methodology: An Overview and Guidelines. *J. Bus. Res.* **2019**, *104*, 333–339. [CrossRef]
17. Tabb, A.; Medeiros, H. A Robotic Vision System to Measure Tree Traits. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 6005–6012. [CrossRef]
18. Tabb, A.; Medeiros, H. Automatic Segmentation of Trees in Dynamic Outdoor Environments. *Comput. Ind.* **2018**, *98*, 90–99. [CrossRef]
19. Svensson, J. Assessment of Grapevine Vigour Using Image Processing. Master's Thesis, Linköping University, Linköping, Sweden, 2002.
20. Chen, Y.; Xiao, K.; Gao, G.; Zhang, F. High-Fidelity 3D Reconstruction of Peach Orchards Using a 3DGS-Ag Model. *Comput. Electron. Agric.* **2025**, *234*, 110225. [CrossRef]
21. Ji, W.; Qian, Z.; Xu, B.; Tao, Y.; Zhao, D.; Ding, S. Apple Tree Branch Segmentation from Images with Small Gray-Level Difference for Agricultural Harvesting Robot. *Optik* **2016**, *127*, 11173–11182. [CrossRef]
22. Ji, W.; Meng, X.; Tao, Y.; Xu, B.; Zhao, D. Fast Segmentation of Colour Apple Image under All-Weather Natural Conditions for Vision Recognition of Picking Robot. *Int. J. Adv. Robot. Syst.* **2017**, *13*, 24. [CrossRef]
23. Silwal, A.; Davidson, J.R.; Karkee, M.; Mo, C.; Zhang, Q.; Lewis, K. Design, Integration, and Field Evaluation of a Robotic Apple Harvester. *J. Field Robot.* **2017**, *34*, 1140–1159. [CrossRef]
24. Xiang, R. Image Segmentation for Whole Tomato Plant Recognition at Night. *Comput. Electron. Agric.* **2018**, *154*, 434–442. [CrossRef]
25. Deng, J.; Li, J.; Zou, X. Extraction of Litchi Stem Based on Computer Vision under Natural Scene. In Proceedings of the International Conference on Computer Distributed Control and Intelligent Environmental Monitoring (CDCIEM), Changsha, China, 19–20 February 2011; pp. 832–835. [CrossRef]
26. Zhuang, J.; Hou, C.; Tang, Y.; He, Y.; Guo, Q.; Zhong, Z.; Luo, S. Computer Vision-Based Localization of Picking Points for Automatic Litchi Harvesting Applications towards Natural Scenarios. *Biosyst. Eng.* **2019**, *187*, 1–20. [CrossRef]
27. Xiong, J.; Lin, R.; Liu, Z.; He, Z.; Tang, L.; Yang, Z.; Zou, X. The Recognition of Litchi Clusters and the Calculation of Picking Point in a Nocturnal Natural Environment. *Biosyst. Eng.* **2018**, *166*, 44–57. [CrossRef]

28. Xiong, J.; He, Z.; Lin, R.; Liu, Z.; Bu, R.; Yang, Z.; Peng, H.; Zou, X. Visual Positioning Technology of Picking Robots for Dynamic Litchi Clusters with Disturbance. *Comput. Electron. Agric.* **2018**, *151*, 226–237. [CrossRef]
29. Pla, F.; Juste, F.; Ferri, F.; Vicens, M. Colour Segmentation Based on a Light Reflection Model to Locate Citrus Fruits for Robotic Harvesting. *Comput. Electron. Agric.* **1993**, *9*, 53–70. [CrossRef]
30. Lü, Q.; Cai, J.; Liu, B.; Deng, L.; Zhang, Y. Identification of Fruit and Branch in Natural Scenes for Citrus Harvesting Robot Using Machine Vision and Support Vector Machine. *Int. J. Agric. Biol. Eng.* **2014**, *7*, 115–121. [CrossRef]
31. Liu, T.-H.; Ehsani, R.; Toudeshki, A.; Zou, X.-J.; Wang, H.-J. Detection of Citrus Fruit and Tree Trunks in Natural Environments Using a Multi-Elliptical Boundary Model. *Comput. Ind.* **2018**, *99*, 9–16. [CrossRef]
32. Amatya, S.; Karkee, M.; Gongal, A.; Zhang, Q.; Whiting, M.D. Detection of Cherry Tree Branches with Full Foliage in Planar Architecture for Automated Sweet-Cherry Harvesting. *Biosyst. Eng.* **2016**, *146*, 3–15. [CrossRef]
33. Amatya, S.; Karkee, M.; Zhang, Q.; Whiting, M.D. Automated Detection of Branch Shaking Locations for Robotic Cherry Harvesting Using Machine Vision. *Robotics* **2017**, *6*, 31. [CrossRef]
34. Mohammadi, P.; Massah, J.; Vakilian, K.A. Robotic Date Fruit Harvesting Using Machine Vision and a 5-DOF Manipulator. *J. Field Robot.* **2023**, *40*, 1408–1423. [CrossRef]
35. He, L.; Du, X.; Qiu, G.; Wu, C. 3D Reconstruction of Chinese Hickory Trees for Mechanical Harvest. In Proceedings of the American Society of Agricultural and Biological Engineers (ASABE) Annual Meeting, Dallas, TX, USA, 29 July–1 August 2012. Paper No. 121340678. [CrossRef]
36. Wu, C.; He, L.; Du, X.; Chen, S. 3D Reconstruction of Chinese Hickory Tree for Dynamics Analysis. *Biosyst. Eng.* **2014**, *119*, 69–79. [CrossRef]
37. Hocevar, M.; Sirok, B.; Jejcic, V.; Godesa, T.; Lesnik, M.; Stajnko, D. Design and Testing of an Automated System for Targeted Spraying in Orchards. *J. Plant Dis. Prot.* **2010**, *117*, 71–79. [CrossRef]
38. Berenstein, R.; Ben-Shahar, O.; Shapiro, A.; Edan, Y. Grape Clusters and Foliage Detection Algorithms for Autonomous Selective Vineyard Sprayer. *Intell. Serv. Robot.* **2010**, *3*, 233–243. [CrossRef]
39. Asaei, H.; Jafari, A.; Loghavi, M. Site-Specific Orchard Sprayer Equipped with Machine Vision for Chemical Usage Management. *Comput. Electron. Agric.* **2019**, *162*, 431–439. [CrossRef]
40. Cheng, Z.; Qi, L.; Cheng, Y. Cherry Tree Crown Extraction from Natural Orchard Images with Complex Backgrounds. *Agriculture* **2021**, *11*, 431. [CrossRef]
41. McFarlane, N.J.B.; Tisseyre, B.; Sinfort, C.; Tillett, R.D.; Sevila, F. Image Analysis for Pruning of Long Wood Grape Vines. *J. Agric. Eng. Res.* **1997**, *66*, 111–119. [CrossRef]
42. Gao, M.; Lu, T.-F. Image Processing and Analysis for Autonomous Grapevine Pruning. In Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA), Luoyang, China, 25–29 June 2006; pp. 922–927. [CrossRef]
43. Roy, P.; Dong, W.; Isler, V. Registering Reconstruction of the Two Sides of Fruit Tree Rows. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1–9. [CrossRef]
44. Dunkan, B.; Bulanon, D.M.; Bulanon, J.I.; Nelson, J. Development of a Cross-Platform Mobile Application for Fruit Yield Estimation. *AgriEngineering* **2024**, *6*, 1807–1826. [CrossRef]
45. Juman, M.A.; Wong, Y.W.; Rajkumar, R.K.; Goh, L.J. A Novel Tree Trunk Detection Method for Oil-Palm Plantation Navigation. *Comput. Electron. Agric.* **2016**, *128*, 172–180. [CrossRef]
46. Zhang, C.; Mouton, C.; Valente, J.; Kooistra, L.; van Ooteghem, R.; de Hoog, D.; van Dalfsen, P.; de Jong, P.F. Automatic Flower Cluster Estimation in Apple Orchards Using Aerial and Ground-Based Point Clouds. *Biosyst. Eng.* **2022**, *221*, 164–180. [CrossRef]
47. Xue, J.; Fan, B.; Yan, J.; Dong, S.; Ding, Q. Trunk Detection Based on Laser Radar and Vision Data Fusion. *Int. J. Agric. Biol. Eng.* **2018**, *11*, 20–26. [CrossRef]
48. Lin, G.; Tang, Y.; Zou, X.; Xiong, J.; Fang, Y. Color-, Depth-, and Shape-Based 3D Fruit Detection. *Precis. Agric.* **2020**, *21*, 1–17. [CrossRef]
49. Xiao, K.; Ma, Y.; Gao, G. An Intelligent Precision Orchard Pesticide Spray Technique Based on the Depth-of-Field Extraction Algorithm. *Comput. Electron. Agric.* **2017**, *133*, 30–36. [CrossRef]
50. Gao, G.; Xiao, K.; Ma, Y. A Leaf-Wall-to-Spray-Device Distance and Leaf-Wall-Density-Based Automatic Route-Planning Spray Algorithm for Vineyards. *Crop Prot.* **2018**, *111*, 33–41. [CrossRef]
51. Gao, G.; Xiao, K.; Jia, Y. A Spraying Path Planning Algorithm Based on Colour-Depth Fusion Segmentation in Peach Orchards. *Comput. Electron. Agric.* **2020**, *173*, 105412. [CrossRef]
52. Gimenez, J.; Sansoni, S.; Tosetti, S.; Capraro, F.; Carelli, R. Trunk Detection in Tree Crops Using RGB-D Images for Structure-Based ICM-SLAM. *Comput. Electron. Agric.* **2022**, *199*, 107099. [CrossRef]
53. Rosell Polo, J.R.; Sanz, R.; Llorens, J.; Arnó, J.; Escolà, A.; Ribes-Dasi, M.; Masip, J.; Camp, F.; Gràcia, F.; Solanelles, F.; et al. A Tractor-Mounted Scanning LiDAR for the Non-Destructive Measurement of Vegetative Volume and Surface Area of Tree-Row Plantations: A Comparison with Conventional Destructive Measurements. *Biosyst. Eng.* **2009**, *102*, 128–134. [CrossRef]

54. Rosell, J.R.; Llorens, J.; Sanz, R.; Arnó, J.; Ribes-Dasi, M.; Masip, J.; Escolà, A.; Camp, F.; Solanelles, F.; Gràcia, F.; et al. Obtaining the Three-Dimensional Structure of Tree Orchards from Remote 2D Terrestrial LiDAR Scanning. *Agric. For. Meteorol.* **2009**, *149*, 1505–1515. [CrossRef]
55. Méndez, V.; Rosell-Polo, J.R.; Sanz, R.; Escolà, A.; Catalán, H. Deciduous Tree Reconstruction Algorithm Based on Cylinder Fitting from Mobile Terrestrial Laser Scanned Point Clouds. *Biosyst. Eng.* **2014**, *124*, 78–88. [CrossRef]
56. Das, J.; Cross, G.; Qu, C.; Makineni, A.; Tokekar, P.; Mulgaonkar, Y.; Kumar, V. Devices, Systems, and Methods for Automated Monitoring Enabling Precision Agriculture. In Proceedings of the IEEE International Conference on Automation Science and Engineering (CASE), Gothenburg, Sweden, 24–28 August 2015; pp. 462–469. [CrossRef]
57. Peng, C.; Roy, P.; Luby, J.; Isler, V. Semantic Mapping of Orchards. *IFAC-Pap. OnLine* **2016**, *49*, 85–89. [CrossRef]
58. Zhang, C.; Yang, G.; Jiang, Y.; Xu, B.; Li, X.; Zhu, Y.; Lei, L.; Chen, R.; Dong, Z.; Yang, H. Apple Tree Branch Information Extraction from Terrestrial Laser Scanning and Backpack-LiDAR. *Remote Sens.* **2020**, *12*, 3592. [CrossRef]
59. Wahabzada, M.; Paulus, S.; Kersting, K.; Mahlein, A.-K. Automated Interpretation of 3D Laser-Scanned Point Clouds for Plant Organ Segmentation. *BMC Bioinform.* **2015**, *16*, 248. [CrossRef]
60. Scholer, F.; Steinhage, V. Automated 3D Reconstruction of Grape Cluster Architecture from Sensor Data for Efficient Phenotyping. *Comput. Electron. Agric.* **2015**, *114*, 163–177. [CrossRef]
61. Mack, J.; Lenz, C.; Teutrine, J.; Steinhage, V. High-Precision 3D Detection and Reconstruction of Grapes from Laser Range Data for Efficient Phenotyping Based on Supervised Learning. *Comput. Electron. Agric.* **2017**, *135*, 300–311. [CrossRef]
62. Zhu, T.; Ma, X.; Guan, H.; Wu, X.; Wang, F.; Yang, C.; Jiang, Q. A Calculation Method of Phenotypic Traits Based on Three-Dimensional Reconstruction of Tomato Canopy. *Comput. Electron. Agric.* **2023**, *204*, 107515. [CrossRef]
63. Zhu, T.; Ma, X.; Guan, H.; Wu, X.; Wang, F.; Yang, C.; Jiang, Q. A Method for Detecting Tomato Canopies' Phenotypic Traits Based on Improved Skeleton Extraction Algorithm. *Comput. Electron. Agric.* **2023**, *214*, 108285. [CrossRef]
64. Auat Cheein, F.A.; Guivant, J.; Sanz, R.; Escolà, A.; Yandún, F.; Torres-Torriti, M.; Rosell-Polo, J.R. Real-Time Approaches for Characterization of Fully and Partially Scanned Canopies in Groves. *Comput. Electron. Agric.* **2015**, *118*, 361–371. [CrossRef]
65. Nielsen, M.; Slaughter, D.C.; Gliever, C.; Upadhyaya, S. Orchard and Tree Mapping and Description Using Stereo Vision and LiDAR. In Proceedings of the International Conference of Agricultural Engineering (AgEng), CIGR–EurAgEng, Valencia, Spain, 8 July 2012.
66. Underwood, J.P.; Jagbrant, G.; Nieto, J.I.; Sukkarieh, S. LiDAR-Based Tree Recognition and Platform Localization in Orchards. *J. Field Robot.* **2015**, *32*, 1056–1074. [CrossRef]
67. Bargoti, S.; Underwood, J.P.; Nieto, J.I.; Sukkarieh, S. A Pipeline for Trunk Detection in Trellis Structured Apple Orchards. *J. Field Robot.* **2015**, *32*, 1075–1094. [CrossRef]
68. Underwood, J.P.; Hung, C.; Whelan, B.; Sukkarieh, S. Mapping Almond Orchard Canopy Volume, Flowers, Fruits and Yield Using LiDAR and Vision Sensors. *Comput. Electron. Agric.* **2016**, *130*, 83–96. [CrossRef]
69. Westling, F.; Underwood, J.; Bryson, M. Graph-Based Methods for Analyzing Orchard Tree Structure Using Noisy Point Cloud Data. *Comput. Electron. Agric.* **2021**, *187*, 106270. [CrossRef]
70. Li, L.; Fu, W.; Zhang, B.; Yang, Y.; Ge, Y.; Shen, C. Branch Segmentation and Phenotype Extraction of Apple Trees Based on Improved Laplace Algorithm. *Comput. Electron. Agric.* **2025**, *232*, 109998. [CrossRef]
71. Tao, Y.; Zhou, J. Automatic Apple Recognition Based on the Fusion of Color and 3D Feature for Robotic Fruit Picking. *Comput. Electron. Agric.* **2017**, *142*, 388–396. [CrossRef]
72. Mahmud, M.S.; Zahid, A.; He, L.; Choi, D.; Krawczyk, G.; Zhu, H.; Heinemann, P. Development of a LiDAR-Guided Section-Based Tree Canopy Density Measurement System for Precision Spray Applications. *Comput. Electron. Agric.* **2021**, *182*, 106053. [CrossRef]
73. Guo, N.; Xu, N.; Kang, J.; Zhang, G.; Meng, Q.; Niu, M.; Wu, W.; Zhang, X. A Study on Canopy Volume Measurement Model for Fruit Tree Application Based on LiDAR Point Cloud. *Agriculture* **2025**, *15*, 130. [CrossRef]
74. Karkee, M.; Adhikari, B.; Amatya, S.; Zhang, Q. Identification of Pruning Branches in Tall Spindle Apple Trees for Automated Pruning. *Comput. Electron. Agric.* **2014**, *103*, 127–135. [CrossRef]
75. Elfiky, N.M.; Akbar, S.A.; Sun, J.; Park, J.; Kak, A. Automation of Dormant Pruning in Specialty Crop Production: An Adaptive Framework for Automatic Reconstruction and Modeling of Apple Trees. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Boston, MA, USA, 7–12 June 2015; pp. 65–73. [CrossRef]
76. Zeng, L.; Feng, J.; He, L. Semantic Segmentation of Sparse 3D Point Cloud Based on Geometrical Features for Trellis-Structured Apple Orchard. *Biosyst. Eng.* **2020**, *196*, 46–55. [CrossRef]
77. You, A.; Grimm, C.; Silwal, A.; Davidson, J.R. Semantics-Guided Skeletonization of Upright Fruiting Offshoot Trees for Robotic Pruning. *Comput. Electron. Agric.* **2022**, *192*, 106622. [CrossRef]
78. You, A.; Parayil, N.; Krishna, J.G.; Bhattarai, U.; Sapkota, R.; Ahmed, D.; Whiting, M.; Karkee, M.; Grimm, C.M.; Davidson, J.R. An Autonomous Robot for Pruning Modern, Planar Fruit Trees. *arXiv* **2022**, arXiv:2206.07201v1. [CrossRef]
79. Li, Y.; Zhang, Z.; Wang, X.; Fu, W.; Li, J. Automatic Reconstruction and Modeling of Dormant Jujube Trees Using Three-View Image Constraints for Intelligent Pruning Applications. *Comput. Electron. Agric.* **2023**, *212*, 108149. [CrossRef]

80. Westling, F.; Underwood, J.; Bryson, M. A Procedure for Automated Tree Pruning Suggestion Using LiDAR Scans of Fruit Trees. *Comput. Electron. Agric.* **2021**, *187*, 106274. [CrossRef]
81. Cao, Y.; Wang, N.; Wu, B.; Zhang, X.; Wang, Y.; Xu, S.; Zhang, M.; Miao, Y.; Kang, F. A Novel Adaptive Cuboid Regional Growth Algorithm for Trunk–Branch Segmentation of Point Clouds from Two Fruit Tree Species. *Agriculture* **2025**, *15*, 1463. [CrossRef]
82. Churuvija, M.; Sapkota, R.; Ahmed, D.; Karkee, M. A Pose-Versatile Imaging System for Comprehensive 3D Modeling of Planar-Canopy Fruit Trees for Automated Orchard Operations. *Comput. Electron. Agric.* **2025**, *230*, 109899. [CrossRef]
83. Zine-El-Abidine, M.; Dutagaci, H.; Galopin, G.; Rousseau, D. Assigning Apples to Individual Trees in Dense Orchards Using 3D Colour Point Clouds. *Biosyst. Eng.* **2021**, *209*, 30–52. [CrossRef]
84. Dey, D.; Mummert, L.; Sukthankar, R. Classification of Plant Structures from Uncalibrated Image Sequences. In Proceedings of the IEEE Workshop on the Applications of Computer Vision (WACV), Breckenridge, CO, USA, 9–11 January 2012; pp. 329–336. [CrossRef]
85. Zhang, J.; Chambers, A.; Maeta, S.; Bergerman, M.; Singh, S. 3D Perception for Accurate Row Following: Methodology and Results. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–7 November 2013; pp. 5306–5313. [CrossRef]
86. Nielsen, M.; Slaughter, D.C.; Gliever, C. Vision-Based 3D Peach Tree Reconstruction for Automated Blossom Thinning. *IEEE Trans. Ind. Inform.* **2012**, *8*, 188–196. [CrossRef]
87. Chéné, Y.; Rousseau, D.; Lucidarme, P.; Bertheloot, J.; Caffier, V.; Morel, P.; Belin, É.; Chapeau-Blondeau, F. On the Use of Depth Camera for 3D Phenotyping of Entire Plants. *Comput. Electron. Agric.* **2012**, *82*, 122–127. [CrossRef]
88. Jin, Y.; Yu, C.; Yin, J.; Yang, S.X. Detection Method for Table Grape Ears and Stems Based on a Far–Close-Range Combined Vision System and Hand–Eye-Coordinated Picking Test. *Comput. Electron. Agric.* **2022**, *202*, 107364. [CrossRef]
89. Lu, Q.; Tang, M.; Cai, J. Obstacle Recognition Using Multi-Spectral Imaging for Citrus Picking Robot. In Proceedings of the Pacific Asia Conference on Circuits, Communications and System (PACCS), Wuhan, China, 17–18 July 2011; pp. 1–5. [CrossRef]
90. Tanigaki, K.; Fujiura, T.; Akase, A.; Imagawa, J. Cherry-Harvesting Robot. *Comput. Electron. Agric.* **2008**, *63*, 65–72. [CrossRef]
91. Bac, C.W.; Hemming, J.; van Henten, E.J. Robust Pixel-Based Classification of Obstacles for Robotic Harvesting of Sweet-Pepper. *Comput. Electron. Agric.* **2013**, *96*, 148–162. [CrossRef]
92. Bac, C.W.; Hemming, J.; van Henten, E.J. Stem Localization of Sweet-Pepper Plants Using the Support Wire as Visual Cue. *Comput. Electron. Agric.* **2014**, *105*, 111–120. [CrossRef]
93. Colmenero-Martinez, J.T.; Blanco-Roldán, G.L.; Bayano-Tejero, S.; Castillo-Ruiz, F.J.; Sola-Guirado, R.R.; Gil-Ribes, J.A. An Automatic Trunk-Detection System for Intensive Olive Harvesting with Trunk Shaker. *Biosyst. Eng.* **2018**, *172*, 92–101. [CrossRef]
94. Chattopadhyay, S.; Akbar, S.A.; Elfiky, N.M.; Medeiros, H.; Kak, A. Measuring and Modeling Apple Trees Using Time-of-Flight Data for Automation of Dormant Pruning Applications. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY, USA, 7–10 March 2016; pp. 1–9. [CrossRef]
95. Akbar, S.A.; Elfiky, N.M.; Kak, A. A Novel Framework for Modeling Dormant Apple Trees Using a Single Depth Image for Robotic Pruning Application. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 5136–5142. [CrossRef]
96. Medeiros, H.; Kim, D.; Sun, J.; Seshadri, H.; Akbar, S.A.; Elfiky, N.M.; Park, J. Modeling Dormant Fruit Trees for Agricultural Automation. *J. Field Robot.* **2017**, *34*, 1203–1224. [CrossRef]
97. Botterill, T.; Green, R.; Mills, S. Finding a Vine's Structure by Bottom-Up Parsing of Cane Edges. In Proceedings of the International Conference on Image and Vision Computing New Zealand (IVCNZ), Wellington, New Zealand, 27–29 November 2013; pp. 112–117. [CrossRef]
98. Botterill, T.; Paulin, S.; Green, R.; Williams, S.; Lin, J.; Saxton, V.; Mills, S.; Chen, X.; Corbett-Davies, S. A Robot System for Pruning Grape Vines. *J. Field Robot.* **2017**, *34*, 1100–1122. [CrossRef]
99. Luo, L.; Tang, Y.; Zou, X.; Ye, M.; Feng, W.; Li, G. Vision-Based Extraction of Spatial Information in Grape Clusters for Harvesting Robots. *Biosyst. Eng.* **2016**, *151*, 90–104. [CrossRef]
100. Gené-Mola, J.; Sanz-Cortiella, R.; Rosell-Polo, J.R.; Morros, J.-R.; Ruiz-Hidalgo, J.; Vilaplana, V.; Gregorio, E. Fruit Detection and 3D Location Using Instance Segmentation Neural Networks and Structure-from-Motion Photogrammetry. *Comput. Electron. Agric.* **2020**, *169*, 105165. [CrossRef]
101. Sun, X.; Fang, W.; Gao, C.; Fu, L.; Majeed, Y.; Liu, X.; Gao, F.; Yang, R.; Li, R. Remote Estimation of Grafted Apple Tree Trunk Diameter in Modern Orchard with RGB and Point Cloud Based on SOLOv2. *Comput. Electron. Agric.* **2022**, *199*, 107209. [CrossRef]
102. Suo, R.; Fu, L.; He, L.; Li, G.; Majeed, Y.; Liu, X.; Zhao, G.; Yang, R.; Li, R. A Novel Labeling Strategy to Improve Apple Seedling Segmentation Using BlendMask for Online Grading. *Comput. Electron. Agric.* **2022**, *201*, 107333. [CrossRef]
103. Zhao, G.; Yang, R.; Jing, X.; Zhang, H.; Wu, Z.; Sun, X.; Jiang, H.; Li, R.; Wei, X.; Fountas, S.; et al. Phenotyping of Individual Apple Tree in Modern Orchard with Novel Smartphone-Based Heterogeneous Binocular Vision and YOLOv5s. *Comput. Electron. Agric.* **2023**, *209*, 107814. [CrossRef]

104. Siddique, A.; Tabb, A.; Medeiros, H. Self-Supervised Learning for Panoptic Segmentation of Multiple Fruit Flower Species. *IEEE Robot. Autom. Lett.* **2022**, *7*, 12387–12394. [CrossRef]
105. Xiong, J.; Liang, J.; Zhuang, Y.; Hong, D.; Zheng, Z.; Liao, S.; Hu, W.; Yang, Z. Real-Time Localization and 3D Semantic Map Reconstruction for Unstructured Citrus Orchards. *Comput. Electron. Agric.* **2023**, *213*, 108217. [CrossRef]
106. Chen, J.; Ji, C.; Zhang, J.; Feng, Q.; Li, Y.; Ma, B. A Method for Multi-Target Segmentation of Bud-Stage Apple Trees Based on Improved YOLOv8. *Comput. Electron. Agric.* **2024**, *220*, 108876. [CrossRef]
107. Xu, W.; Guo, R.; Chen, P.; Li, L.; Gu, M.; Sun, H.; Hu, L.; Wang, Z.; Li, K. Cherry Growth Modeling Based on Prior Distance Embedding Contrastive Learning: Pre-Training, Anomaly Detection, Semantic Segmentation, and Temporal Modeling. *Comput. Electron. Agric.* **2024**, *221*, 108973. [CrossRef]
108. Zhang, J.; Tian, M.; Yang, Z.; Li, J.; Zhao, L. An Improved Target Detection Method Based on YOLOv5 in Natural Orchard Environments. *Comput. Electron. Agric.* **2024**, *219*, 108780. [CrossRef]
109. Abdalla, A.; Zhu, Y.; Qu, F.; Cen, H. Novel Encoding Technique to Evolve Convolutional Neural Network as a Multi-Criteria Problem for Plant Image Segmentation. *Comput. Electron. Agric.* **2025**, *230*, 109869. [CrossRef]
110. Gao, G.; Fang, L.; Zhang, Z.; Li, J. YOLOR-Stem: Gaussian Rotating Bounding Boxes and Probability Similarity Measure for Enhanced Tomato Main Stem Detection. *Comput. Electron. Agric.* **2025**, *233*, 110192. [CrossRef]
111. Jin, T.; Kang, S.M.; Kim, N.; Kim, H.R.; Han, X. Comparative Analysis of CNN-Based Semantic Segmentation for Apple Tree Canopy Size Recognition in Automated Variable-Rate Spraying. *Agriculture* **2025**, *15*, 789. [CrossRef]
112. Metuarea, H.; Laurens, F.; Guerra, W.; Lozano, L.; Patocchi, A.; Van Hoye, S.; Dutagaci, H.; Labrosse, J.; Rasti, P.; Rousseau, D. Individual Segmentation of Intertwined Apple Trees in a Row via Prompt Engineering. *Sensors* **2025**, *25*, 4721. [CrossRef]
113. Kang, H.; Chen, C. Fruit Detection and Segmentation for Apple Harvesting Using Visual Sensor in Orchards. *Sensors* **2019**, *19*, 4599. [CrossRef]
114. Kang, H.; Chen, C. Fruit Detection, Segmentation and 3D Visualization of Environments in Apple Orchards. *Comput. Electron. Agric.* **2020**, *171*, 105302. [CrossRef]
115. Jiang, H.; Sun, X.; Fang, W.; Fu, L.; Li, R.; Auat Cheein, F.; Majeed, Y. Thin Wire Segmentation and Reconstruction Based on a Novel Image Overlap-Partitioning and Stitching Algorithm in Apple Fruiting Wall Architecture for Robotic Picking. *Comput. Electron. Agric.* **2023**, *209*, 107840. [CrossRef]
116. Kok, E.; Wang, X.; Chen, C. Obscured Tree Branches Segmentation and 3D Reconstruction Using Deep Learning and Geometrical Constraints. *Comput. Electron. Agric.* **2023**, *210*, 107884. [CrossRef]
117. Kalampokas, T.; Tziridis, K.; Nikolaou, A.; Vrochidou, E.; Papakostas, G.A.; Pachidis, T.; Kaburlasos, V.G. Semantic Segmentation of Vineyard Images Using Convolutional Neural Networks. In *International Neural Networks Society, Proceedings of the International Conference on Engineering Applications of Neural Networks (EANN), Halkidiki, Greece, 5–7 June 2020*; Springer: Berlin/Heidelberg, Germany; p. 2. [CrossRef]
118. Kalampokas, T.; Vrochidou, E.; Papakostas, G.A.; Pachidis, T.; Kaburlasos, V.G. Grape Stem Detection Using Regression Convolutional Neural Networks. *Comput. Electron. Agric.* **2021**, *186*, 106220. [CrossRef]
119. Qiao, Y.; Hu, Y.; Zheng, Z.; Qu, Z.; Wang, C.; Guo, T.; Hou, J. A Diameter Measurement Method of Red Jujubes Trunk Based on Improved PSPNet. *Agriculture* **2022**, *12*, 1140. [CrossRef]
120. Wang, J.; Zhang, Z.; Luo, L.; Wei, H.; Wang, W.; Chen, M.; Luo, S. DualSeg: Fusing Transformer and CNN Structure for Image Segmentation in Complex Vineyard Environment. *Comput. Electron. Agric.* **2023**, *206*, 107682. [CrossRef]
121. Wu, Z.; Xia, F.; Zhou, S.; Xu, D. A Method for Identifying Grape Stems Using Keypoints. *Comput. Electron. Agric.* **2023**, *209*, 107825. [CrossRef]
122. Kim, J.; Pyo, H.; Jang, I.; Kang, J.; Ju, B.; Ko, K. Tomato Harvesting Robotic System Based on Deep-ToMaToS: Deep Learning Network Using Transformation Loss for 6D Pose Estimation of Maturity Classified Tomatoes with Side-Stem. *Comput. Electron. Agric.* **2022**, *201*, 107300. [CrossRef]
123. Rong, J.; Wang, P.; Wang, T.; Hu, L.; Yuan, T. Fruit Pose Recognition and Directional Orderly Grasping Strategies for Tomato Harvesting Robots. *Comput. Electron. Agric.* **2022**, *202*, 107430. [CrossRef]
124. Rong, Q.; Hu, C.; Hu, X.; Xu, M. Picking Point Recognition for Ripe Tomatoes Using Semantic Segmentation and Morphological Processing. *Comput. Electron. Agric.* **2023**, *210*, 107923. [CrossRef]
125. Kim, T.; Lee, D.-H.; Kim, K.-C.; Kim, Y.-J. 2D Pose Estimation of Multiple Tomato Fruit-Bearing Systems for Robotic Harvesting. *Comput. Electron. Agric.* **2023**, *211*, 108004. [CrossRef]
126. Liang, C.; Xiong, J.; Zheng, Z.; Zhong, Z.; Li, Z.; Chen, S.; Yang, Z. A Visual Detection Method for Nighttime Litchi Fruits and Fruiting Stems. *Comput. Electron. Agric.* **2020**, *169*, 105192. [CrossRef]
127. Chen, M.; Tang, Y.; Zou, X.; Huang, Z.; Zhou, H.; Chen, S. 3D Global Mapping of Large-Scale Unstructured Orchard Integrating Eye-in-Hand Stereo Vision and SLAM. *Comput. Electron. Agric.* **2021**, *187*, 106237. [CrossRef]
128. Zhong, Z.; Xiong, J.; Zheng, Z.; Liu, B.; Liao, S.; Huo, Z.; Yang, Z. A Method for Litchi Picking Points Calculation in Natural Environment Based on Main Fruit Bearing Branch Detection. *Comput. Electron. Agric.* **2021**, *189*, 106398. [CrossRef]

129. Peng, H.; Zhong, J.; Liu, H.; Li, J.; Yao, M.; Zhang, X. ResDense-Focal-DeepLabV3+ Enabled Litchi Branch Semantic Segmentation for Robotic Harvesting. *Comput. Electron. Agric.* **2023**, *206*, 107691. [CrossRef]
130. Lin, G.; Wang, C.; Xu, Y.; Wang, M.; Zhang, Z.; Zhu, L. Real-Time Guava Tree-Part Segmentation Using Fully Convolutional Network with Channel and Spatial Attention. *Front. Plant Sci.* **2022**, *13*, 991487. [CrossRef] [PubMed]
131. Wang, Y.; Deng, X.; Luo, J.; Li, B.; Xiao, S. Cross-Task Feature Enhancement Strategy in Multi-Task Learning for Harvesting Sichuan Pepper. *Comput. Electron. Agric.* **2023**, *207*, 107726. [CrossRef]
132. Zheng, Z.; Hu, Y.; Guo, T.; Qiao, Y.; He, Y.; Zhang, Y.; Huang, Y. AGHRNet: An Attention Ghost-HRNet for Confirmation of Catch-and-Shake Locations in Jujube Fruits Vibration Harvesting. *Comput. Electron. Agric.* **2023**, *210*, 107921. [CrossRef]
133. Williams, H.A.M.; Jones, M.H.; Nejati, M.; Seabright, M.J.; Bell, J.; Penhall, N.D.; Barnett, J.J.; Duke, M.D.; Scarfe, A.J.; Ahn, H.S.; et al. Robotic Kiwifruit Harvesting Using Machine Vision, Convolutional Neural Networks, and Robotic Arms. *Biosyst. Eng.* **2019**, *181*, 140–156. [CrossRef]
134. Williams, H.; Ting, C.; Nejati, M.; Jones, M.H.; Penhall, N.; Lim, J.; Seabright, M.; Bell, J.; Ahn, H.S.; Scarfe, A.; et al. Improvements to and Large-Scale Evaluation of a Robotic Kiwifruit Harvester. *J. Field Robot.* **2020**, *37*, 187–201. [CrossRef]
135. Song, Z.; Zhou, Z.; Wang, W.; Gao, F.; Fu, L.; Li, R.; Cui, Y. Canopy Segmentation and Wire Reconstruction for Kiwifruit Robotic Harvesting. *Comput. Electron. Agric.* **2021**, *181*, 105933. [CrossRef]
136. Fu, L.; Wu, F.; Zou, X.; Jiang, Y.; Lin, J.; Yang, Z.; Duan, J. Fast Detection of Banana Bunches and Stalks in the Natural Environment Based on Deep Learning. *Comput. Electron. Agric.* **2022**, *194*, 106800. [CrossRef]
137. Wan, H.; Fan, Z.; Yu, X.; Kang, M.; Wang, P.; Zeng, X. A Real-Time Branch Detection and Reconstruction Mechanism for Harvesting Robot via Convolutional Neural Network and Image Segmentation. *Comput. Electron. Agric.* **2022**, *192*, 106609. [CrossRef]
138. Li, D.; Sun, X.; Lv, S.; Elkhouchlaa, H.; Jia, Y.; Yao, Z.; Lin, P.; Zhou, H.; Zhou, Z.; Shen, J.; et al. A Novel Approach for the 3D Localization of Branch Picking Points Based on Deep Learning Applied to Longan Harvesting UAVs. *Comput. Electron. Agric.* **2022**, *199*, 107191. [CrossRef]
139. Chen, J.; Ma, A.; Huang, L.; Li, H.; Zhang, H.; Huang, Y.; Zhu, T. Efficient and Lightweight Grape and Picking Point Synchronous Detection Model Based on Key Point Detection. *Comput. Electron. Agric.* **2024**, *217*, 108612. [CrossRef]
140. Du, X.; Meng, Z.; Ma, Z.; Zhao, L.; Lu, W.; Cheng, H.; Wang, Y. Comprehensive Visual Information Acquisition for Tomato Picking Robot Based on Multitask Convolutional Neural Network. *Biosyst. Eng.* **2024**, *238*, 51–61. [CrossRef]
141. Gu, Z.; He, D.; Huang, J.; Chen, J.; Wu, X.; Huang, B.; Dong, T.; Yang, Q.; Li, H. Simultaneous Detection of Fruits and Fruiting Stems in Mango Using Improved YOLOv8 Model Deployed by Edge Device. *Comput. Electron. Agric.* **2024**, *227*, 109512. [CrossRef]
142. Li, H.; Huang, J.; Gu, Z.; He, D.; Huang, J.; Wang, C. Positioning of Mango Picking Point Using an Improved YOLOv8 Architecture with Object Detection and Instance Segmentation. *Biosyst. Eng.* **2024**, *247*, 202–220. [CrossRef]
143. Neupane, C.; Walsh, K.B.; Goulart, R.; Koirala, A. Developing Machine Vision in Tree-Fruit Applications: Fruit Count, Fruit Size and Branch Avoidance in Automated Harvesting. *Sensors* **2024**, *24*, 5593. [CrossRef]
144. Sapkota, R.; Ahmed, D.; Karkee, M. Comparing YOLOv8 and Mask R-CNN for Instance Segmentation in Complex Orchard Environments. *Artif. Intell. Agric.* **2024**, *13*, 84–99. [CrossRef]
145. Wang, J.; Lin, X.; Luo, L.; Chen, M.; Wei, H.; Xu, L.; Luo, S. Cognition of Grape Cluster Picking Point Based on Visual Knowledge Distillation in Complex Vineyard Environment. *Comput. Electron. Agric.* **2024**, *225*, 109216. [CrossRef]
146. Wang, C.; Han, Q.; Zhang, T.; Li, C.; Sun, X. Litchi Picking Points Localization in Natural Environment Based on the Litchi-YOSO Model and Branch Morphology Reconstruction Algorithm. *Comput. Electron. Agric.* **2024**, *226*, 109473. [CrossRef]
147. Li, L.; Li, K.; He, Z.; Li, H.; Cui, Y. Kiwifruit Segmentation and Identification of Picking Point on Its Stem in Orchards. *Comput. Electron. Agric.* **2025**, *229*, 109748. [CrossRef]
148. Li, P.; Chen, J.; Chen, Q.; Huang, L.; Jiang, Z.; Hua, W.; Li, Y. Detection and Picking Point Localization of Grape Bunches and Stems Based on Oriented Bounding Box. *Comput. Electron. Agric.* **2025**, *233*, 110168. [CrossRef]
149. Shen, Q.; Zhang, X.; Shen, M.; Xu, D. Multi-Scale Adaptive YOLO for Instance Segmentation of Grape Pedicels. *Comput. Electron. Agric.* **2025**, *229*, 109712. [CrossRef]
150. Wu, Y.; Yu, X.; Zhang, D.; Yang, Y.; Qiu, Y.; Pang, L.; Wang, H. TinySeg: A Deep Learning Model for Small Target Segmentation of Grape Pedicels with Multi-Attention and Multi-Scale Feature Fusion. *Comput. Electron. Agric.* **2025**, *237*, 110726. [CrossRef]
151. Kim, J.; Seol, J.; Lee, S.; Hong, S.-W.; Son, H.I. An Intelligent Spraying System with Deep Learning-Based Semantic Segmentation of Fruit Trees in Orchards. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 3923–3929. [CrossRef]
152. Seol, J.; Kim, J.; Son, H.I. Field Evaluation of a Deep Learning-Based Intelligent Spraying Robot with Flow Control for Pear Orchards. *Precis. Agric.* **2022**, *23*, 712–732. [CrossRef]
153. Tong, S.; Yue, Y.; Li, W.; Wang, Y.; Kang, F.; Feng, C. Branch Identification and Junction Points Location for Apple Trees Based on Deep Learning. *Remote Sens.* **2022**, *14*, 4495. [CrossRef]

154. Tong, S.; Zhang, J.; Li, W.; Wang, Y.; Kang, F. An Image-Based System for Locating Pruning Points in Apple Trees Using Instance Segmentation and RGB-D Images. *Biosyst. Eng.* **2023**, *236*, 277–286. [CrossRef]
155. Williams, H.; Smith, D.; Shahabi, J.; Gee, T.; Nejati, M.; McGuinness, B.; Black, K.; Tobias, J.; Jangali, R.; Lim, H.; et al. Modelling Wine Grapevines for Autonomous Robotic Cane Pruning. *Biosyst. Eng.* **2023**, *235*, 31–49. [CrossRef]
156. Gentilhomme, T.; Villamizar, M.; Corre, J.; Odobez, J.-M. Towards Smart Pruning: ViNet, a Deep-Learning Approach for Grapevine Structure Estimation. *Comput. Electron. Agric.* **2023**, *207*, 107736. [CrossRef]
157. Liang, X.; Wei, Z.; Chen, K. A Method for Segmentation and Localization of Tomato Lateral Pruning Points in Complex Environments Based on Improved YOLOv5. *Comput. Electron. Agric.* **2025**, *229*, 109731. [CrossRef]
158. Hani, N.; Roy, P.; Isler, V. A Comparative Study of Fruit Detection and Counting Methods for Yield Mapping in Apple Orchards. *J. Field Robot.* **2020**, *37*, 263–282. [CrossRef]
159. Gao, F.; Fang, W.; Sun, X.; Wu, Z.; Zhao, G.; Li, G.; Li, R.; Fu, L.; Zhang, Q. A Novel Apple Fruit Detection and Counting Methodology Based on Deep Learning and Trunk Tracking in Modern Orchard. *Comput. Electron. Agric.* **2022**, *197*, 107000. [CrossRef]
160. Wu, Z.; Sun, X.; Jiang, H.; Gao, F.; Li, R.; Fu, L.; Zhang, D.; Fountas, S. Twice Matched Fruit Counting System: An Automatic Fruit Counting Pipeline in Modern Apple Orchard Using Mutual and Secondary Matches. *Biosyst. Eng.* **2023**, *234*, 140–155. [CrossRef]
161. Palacios, F.; Melo-Pinto, P.; Diago, M.P.; Tardaguila, J. Deep Learning and Computer Vision for Assessing the Number of Actual Berries in Commercial Vineyards. *Biosyst. Eng.* **2022**, *218*, 175–188. [CrossRef]
162. Wen, Y.; Xue, J.; Sun, H.; Song, Y.; Lv, P.; Liu, S.; Chu, Y.; Zhang, T. High-Precision Target Ranging in Complex Orchard Scenes by Utilizing Semantic Segmentation Results and Binocular Vision. *Comput. Electron. Agric.* **2023**, *215*, 108440. [CrossRef]
163. Saha, S.; Noguchi, N. Smart Vineyard Row Navigation: A Machine Vision Approach Leveraging YOLOv8. *Comput. Electron. Agric.* **2025**, *229*, 109839. [CrossRef]
164. Majeed, Y.; Karkee, M.; Zhang, Q.; Fu, L.; Whiting, M.D. Determining Grapevine Cordon Shape for Automated Green Shoot Thinning Using Semantic Segmentation-Based Deep Learning Networks. *Comput. Electron. Agric.* **2020**, *171*, 105308. [CrossRef]
165. Majeed, Y.; Karkee, M.; Zhang, Q. Estimating the Trajectories of Vine Cordons in Full Foliage Canopies for Automated Green Shoot Thinning in Vineyards. *Comput. Electron. Agric.* **2020**, *176*, 105671. [CrossRef]
166. Wu, F.; Duan, J.; Ai, P.; Chen, Z.; Yang, Z.; Zou, X. Rachis Detection and Three-Dimensional Localization of Cut Point for Vision-Based Banana Robot. *Comput. Electron. Agric.* **2022**, *198*, 107079. [CrossRef]
167. Du, W.; Cui, X.; Zhu, Y.; Liu, P. Detection of Table Grape Berries Need to be Removal before Thinning Based on Deep Learning. *Comput. Electron. Agric.* **2025**, *231*, 110043. [CrossRef]
168. Hussain, M.; He, L.; Schupp, J.; Lyons, D.; Heinemann, P. Green Fruit–Stem Pairing and Clustering for Machine Vision System in Robotic Thinning of Apples. *J. Field Robot.* **2025**, *42*, 1463–1490. [CrossRef]
169. Dong, W.; Roy, P.; Isler, V. Semantic Mapping for Orchard Environments by Merging Two-Sides Reconstructions of Tree Rows. *J. Field Robot.* **2020**, *37*, 97–121. [CrossRef]
170. Milella, A.; Marani, R.; Petitti, A.; Reina, G. In-Field High Throughput Grapevine Phenotyping with a Consumer-Grade Depth Camera. *Comput. Electron. Agric.* **2019**, *156*, 293–306. [CrossRef]
171. Digumarti, S.T.; Schmid, L.M.; Rizzi, G.M.; Nieto, J.; Siegwart, R.; Beardsley, P.; Cadena, C. An Approach for Semantic Segmentation of Tree-like Vegetation. In Proceedings of the 2019 IEEE International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 1801–1807. [CrossRef]
172. Barth, R.; Ijsselmuiden, J.; Hemming, J.; van Henten, E.J. Data Synthesis Methods for Semantic Segmentation in Agriculture: A *Capsicum annuum* Dataset. *Comput. Electron. Agric.* **2018**, *144*, 284–296. [CrossRef]
173. Barth, R.; Ijsselmuiden, J.; Hemming, J.; van Henten, E.J. Synthetic Bootstrapping of Convolutional Neural Networks for Semantic Plant Part Segmentation. *Comput. Electron. Agric.* **2019**, *161*, 291–304. [CrossRef]
174. Barth, R.; Hemming, J.; van Henten, E.J. Optimizing Realism of Synthetic Images Using Cycle Generative Adversarial Networks for Improved Part Segmentation. *Comput. Electron. Agric.* **2020**, *173*, 105378. [CrossRef]
175. Chen, Z.; Granland, K.; Tang, Y.; Chen, C. HOB-CNNv2: Deep Learning Based Detection of Extremely Occluded Tree Branches and Reference to the Dominant Tree Image. *Comput. Electron. Agric.* **2024**, *218*, 108727. [CrossRef]
176. Qi, Z.; Hua, W.; Zhang, Z.; Deng, X.; Yuan, T.; Zhang, W. A Novel Method for Tomato Stem Diameter Measurement Based on Improved YOLOv8-Seg and RGB-D Data. *Comput. Electron. Agric.* **2024**, *226*, 109387. [CrossRef]
177. Zhang, J.; He, L.; Karkee, M.; Zhang, Q.; Zhang, X.; Gao, Z. Branch Detection for Apple Trees Trained in Fruiting Wall Architecture Using Depth Features and Region-Convolutional Neural Network (R-CNN). *Comput. Electron. Agric.* **2018**, *155*, 386–393. [CrossRef]
178. Zhang, X.; Fu, L.; Karkee, M.; Whiting, M.D.; Zhang, Q. Canopy Segmentation Using ResNet for Mechanical Harvesting of Apples. *IFAC-PapersOnLine* **2019**, *52*, 300–305. [CrossRef]
179. Zhang, X.; Karkee, M.; Zhang, Q.; Whiting, M.D. Computer Vision-Based Tree Trunk and Branch Identification and Shaking Point Detection in Dense-Foliage Canopy for Automated Harvesting of Apples. *J. Field Robot.* **2020**, *38*, 476–493. [CrossRef]

180. Zhang, J.; Karkee, M.; Zhang, Q.; Zhang, X.; Majeed, Y.; Fu, L.; Wang, S. Multi-Class Object Detection Using Faster R-CNN and Estimation of Shaking Locations for Automated Shake-and-Catch Apple Harvesting. *Comput. Electron. Agric.* **2020**, *173*, 105384. [CrossRef]
181. Granland, K.; Newbury, R.; Chen, Z.; Ting, D.; Chen, C. Detecting Occluded Y-Shaped Fruit Tree Segments Using Automated Iterative Training with Minimal Labeling Effort. *Comput. Electron. Agric.* **2022**, *194*, 106747. [CrossRef]
182. Coll-Ribes, G.; Torres-Rodriguez, I.J.; Grau, A.; Guerra, E.; Safeliu, A. Accurate Detection and Depth Estimation of Table Grapes and Peduncles for Robot Harvesting, Combining Monocular Depth Estimation and CNN Methods. *Comput. Electron. Agric.* **2023**, *215*, 108362. [CrossRef]
183. Xu, P.; Fang, N.; Liu, N.; Lin, F.; Yang, S.; Ning, J. Visual Recognition of Cherry Tomatoes in Plant Factory Based on Improved Deep Instance Segmentation. *Comput. Electron. Agric.* **2022**, *197*, 106991. [CrossRef]
184. Zhang, F.; Gao, J.; Zhou, H.; Zhang, J.; Zou, K.; Yuan, T. Three-Dimensional Pose Detection Method Based on Keypoints Detection Network for Tomato Bunch. *Comput. Electron. Agric.* **2022**, *195*, 106824. [CrossRef]
185. Zhang, F.; Gao, J.; Song, C.; Zhou, H.; Zou, K.; Xie, J.; Yuan, T.; Zhang, J. TPMv2: An End-to-End Tomato Pose Method Based on 3D Keypoint Detection. *Comput. Electron. Agric.* **2023**, *210*, 107878. [CrossRef]
186. Li, J.; Tang, Y.; Zou, X.; Lin, G.; Wang, H. Detection of Fruit-Bearing Branches and Localization of Litchi Clusters for Vision-Based Harvesting Robots. *IEEE Access* **2020**, *8*, 117746–117758. [CrossRef]
187. Yang, C.H.; Xiong, L.Y.; Wang, Z.; Wang, Y.; Shi, G.; Kuremot, T.; Zhao, W.H.; Yang, Y. Integrated Detection of Citrus Fruits and Branches Using a Convolutional Neural Network. *Comput. Electron. Agric.* **2020**, *174*, 105469. [CrossRef]
188. Lin, G.; Tang, Y.; Zou, X.; Xiong, J.; Li, J. Guava Detection and Pose Estimation Using a Low-Cost RGB-D Sensor in the Field. *Sensors* **2019**, *19*, 428. [CrossRef]
189. Lin, G.; Tang, Y.; Zou, X.; Wang, C. Three-Dimensional Reconstruction of Guava Fruit and Branches Using Instance Segmentation and Geometry Analysis. *Comput. Electron. Agric.* **2021**, *184*, 106107. [CrossRef]
190. Lin, G.; Zhu, L.; Li, J.; Zou, X.; Tang, Y. Collision-Free Path Planning for a Guava-Harvesting Robot Based on Recurrent Deep Reinforcement Learning. *Comput. Electron. Agric.* **2021**, *188*, 106350. [CrossRef]
191. Yu, T.; Hu, C.; Xie, Y.; Liu, J.; Li, P. Mature Pomegranate Fruit Detection and Location Combining Improved F-PointNet with 3D Point Cloud Clustering in Orchard. *Comput. Electron. Agric.* **2022**, *200*, 107233. [CrossRef]
192. Ci, J.; Wang, X.; Rapado-Rincón, D.; Burusa, A.K.; Kootstra, G. 3D Pose Estimation of Tomato Peduncle Nodes Using Deep Keypoint Detection and Point Cloud. *Biosyst. Eng.* **2024**, *243*, 57–69. [CrossRef]
193. Li, Y.; Feng, Q.; Zhang, Y.; Peng, C.; Ma, Y.; Liu, C.; Ru, M.; Sun, J.; Zhao, C. Peduncle Collision-Free Grasping Based on Deep Reinforcement Learning for Tomato Harvesting Robot. *Comput. Electron. Agric.* **2024**, *216*, 108488. [CrossRef]
194. Dong, L.; Zhu, L.; Zhao, B.; Wang, R.; Ni, J.; Liu, S.; Chen, K.; Cui, X.; Zhou, L. Semantic Segmentation-Based Observation Pose Estimation Method for Tomato Harvesting Robots. *Comput. Electron. Agric.* **2025**, *230*, 109895. [CrossRef]
195. Cong, P.; Zhou, J.; Li, S.; Lv, K.; Feng, H. Citrus Tree Crown Segmentation of Orchard Spraying Robot Based on RGB-D Image and Improved Mask R-CNN. *Appl. Sci.* **2023**, *13*, 164. [CrossRef]
196. Bhattarai, U.; Sapkota, R.; Kshetri, S.; Mo, C.; Whiting, M.D.; Zhang, Q.; Karkee, M. A Vision-Based Robotic System for Precision Pollination of Apples. *Comput. Electron. Agric.* **2025**, *234*, 110158. [CrossRef]
197. Chen, Z.; Ting, D.; Newbury, R.; Chen, C. Semantic Segmentation for Partially Occluded Apple Trees Based on Deep Learning. *Comput. Electron. Agric.* **2021**, *181*, 105952. [CrossRef]
198. Ahmed, D.; Sapkota, R.; Churuvija, M.; Karkee, M. Estimating Optimal Crop-Load for Individual Branches in Apple Tree Canopies Using YOLOv8. *Comput. Electron. Agric.* **2025**, *229*, 109697. [CrossRef]
199. Majeed, Y.; Zhang, J.; Zhang, X.; Fu, L.; Karkee, M.; Zhang, Q.; Whiting, M.D. Deep Learning Based Segmentation for Automated Training of Apple Trees on Trellis Wires. *Comput. Electron. Agric.* **2020**, *170*, 105277. [CrossRef]
200. Brown, J.; Paudel, A.; Biehler, D.; Thompson, A.; Karkee, M.; Grimm, C.; Davidson, J.R. Tree Detection and In-Row Localization for Autonomous Precision Orchard Management. *Comput. Electron. Agric.* **2024**, *227*, 109454. [CrossRef]
201. Xu, S.; Rai, R. Vision-Based Autonomous Navigation Stack for Tractors Operating in Peach Orchards. *Comput. Electron. Agric.* **2024**, *217*, 108558. [CrossRef]
202. Pawikhum, K.; Yang, Y.; He, L.; Heinemann, P. Development of a Machine Vision System for Apple Bud Thinning in Precision Crop Load Management. *Comput. Electron. Agric.* **2025**, *236*, 110479. [CrossRef]
203. Dong, X.; Kim, W.-Y.; Zheng, Y.; Oh, J.-Y.; Ehsani, R.; Lee, K.-H. Three-Dimensional Quantification of Apple Phenotypic Traits Based on Deep Learning Instance Segmentation. *Comput. Electron. Agric.* **2023**, *212*, 108156. [CrossRef]
204. Schunck, D.; Magistri, F.; Rosu, R.A.; Cornelißen, A.; Chebrolu, N.; Paulus, S.; Léon, J.; Behnke, S.; Stachniss, C.; Kuhlmann, H.; et al. Pheno4D: A Spatio-Temporal Datasets of Maize and Tomato Plant Point Clouds for Phenotyping and Advanced Plant Analysis. *PLoS ONE* **2021**, *16*, e0256340. [CrossRef] [PubMed]
205. Liu, Q.; Yang, H.; Wei, J.; Zhang, Y.; Yang, S. FSDNet: A Feature Spreading Network with Density for 3D Segmentation in Agriculture. *Comput. Electron. Agric.* **2024**, *222*, 109073. [CrossRef]

206. Sun, X.; He, L.; Jiang, H.; Li, R.; Mao, W.; Zhang, D.; Majeed, Y.; Andriyanov, N.; Soloviev, V.; Fu, L. Morphological Estimation of Primary Branch Length of Individual Apple Trees during the Deciduous Period in Modern Orchard Based on PointNet++. *Comput. Electron. Agric.* **2024**, *220*, 108873. [CrossRef]
207. Jiang, L.; Li, C.; Fu, L. Apple Tree Architectural Trait Phenotyping with Organ-Level Instance Segmentation from Point Cloud. *Comput. Electron. Agric.* **2025**, *229*, 109708. [CrossRef]
208. Luo, L.; Yin, W.; Ning, Z.; Wang, J.; Wei, H.; Chen, W.; Lu, Q. In-Field Pose Estimation of Grape Clusters with Combined Point Cloud Segmentation and Geometric Analysis. *Comput. Electron. Agric.* **2022**, *200*, 107197. [CrossRef]
209. Ma, B.; Du, J.; Wang, L.; Jiang, H.; Zhou, M. Automatic Branch Detection of Jujube Trees Based on 3D Reconstruction for Dormant Pruning Using a Deep Learning-Based Method. *Comput. Electron. Agric.* **2021**, *190*, 106484. [CrossRef]
210. Zhang, J.; Gu, J.; Hu, T.; Wang, B.; Xia, Z. An Image Segmentation and Point Cloud Registration Combined Scheme for Sensing of Obscured Tree Branches. *Comput. Electron. Agric.* **2024**, *221*, 108960. [CrossRef]
211. Zhao, G.; Wang, D. A Multiple Criteria Decision-Making Method Generated by the Space Colonization Algorithms for Automated Pruning Strategies of Trees. *AgriEngineering* **2024**, *6*, 539–554. [CrossRef]
212. Fernandes, M.; Gamba, J.D.; Pelusi, F.; Bratta, A.; Caldwell, D.; Poni, S.; Gatti, M.; Semini, C. Grapevine Winter Pruning: Merging 2D Segmentation and 3D Point Clouds for Pruning Point Generation. *Comput. Electron. Agric.* **2025**, *237*, 110589. [CrossRef]
213. Shang, L.; Yan, F.; Teng, T.; Pan, J.; Zhou, L.; Xia, C.; Li, C.; Shi, M.; Si, C.; Niu, R. Morphological Estimation of Primary Branch Inclination Angles in Jujube Trees Based on Improved PointNet++. *Agriculture* **2025**, *15*, 1193. [CrossRef]
214. Zhu, W.; Bai, X.; Xu, D.; Li, W. Pruning Branch Recognition and Pruning Point Localization for Walnut (*Juglans regia* L.) Trees Based on Point Cloud Semantic Segmentation. *Agriculture* **2025**, *15*, 817. [CrossRef]
215. Uryasheva, A.; Kalashnikova, A.; Shadrin, D.; Evteeva, K.; Moskovtsev, E.; Rodichenko, N. Computer Vision-Based Platform for Apple Leaf Segmentation in Field Conditions to Support Digital Phenotyping. *Comput. Electron. Agric.* **2022**, *201*, 107269. [CrossRef]
216. Liu, C.; Feng, Q.; Sun, Y.; Li, Y.; Ru, M.; Xu, L. YOLACTFusion: An Instance Segmentation Method for RGB–NIR Multimodal Image Fusion Based on an Attention Mechanism. *Comput. Electron. Agric.* **2023**, *213*, 108186. [CrossRef]
217. Hung, C.; Nieto, J.; Taylor, Z.; Underwood, J.; Sukkarieh, S. Orchard Fruit Segmentation Using Multispectral Feature Learning. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 5314–5320. [CrossRef]
218. Droukas, L.; Doulgeri, Z.; Tsakiridis, N.L.; Triantafyllou, D.; Kleitsiotis, I.; Mariolis, I.; Giakoumis, D.; Tzovaras, D.; Kateris, D.; Bochtis, D. A Survey of Robotic Harvesting Systems and Enabling Technologies. *J. Intell. Robot. Syst.* **2023**, *107*, 21. [CrossRef]
219. Westling, F.; Bryson, M.; Underwood, J. SimTreeLS: Simulating Aerial and Terrestrial Laser Scans of Trees. *Comput. Electron. Agric.* **2021**, *187*, 106277. [CrossRef]
220. Scarfe, A.J.; Flemmer, R.C.; Bakker, H.H.; Flemmer, C.L. Development of an Autonomous Kiwifruit Picking Robot. In Proceedings of the 4th International Conference on Autonomous Robots and Agents, Wellington, New Zealand, 10–12 February 2009; pp. 380–384. [CrossRef]
221. Kayad, A.; Sozzi, M.; Paraforos, D.S.; Rodrigues, F.A.; Cohen, Y.; Fountas, S.; Francisco, M.-J.; Pezzuolo, A.; Grigolato, S.; Marinello, F. How Many Gigabytes per Hectare Are Available in the Digital Agriculture Era? A Digitization Footprint Estimation. *Comput. Electron. Agric.* **2022**, *198*, 107080. [CrossRef]
222. Zhu, X.; Zhu, J.; Li, H.; Wu, X.; Li, H.; Wang, X.; Dai, J. Uni-Perceiver: Pre-Training Unified Architecture for Generic Perception for Zero-Shot and Few-Shot Tasks. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 16783–16794. [CrossRef]
223. Li, J.; Zhu, G.; Hua, C.; Feng, M.; Bennamoun, B.; Li, P.; Lu, X.; Song, J.; Shen, P.; Xu, X.; et al. A Systematic Collection of Medical Image Datasets for Deep Learning. *ACM Comput. Surv.* **2023**, *56*, 1–51. [CrossRef]
224. Lu, Y.; Young, S. A Survey of Public Datasets for Computer Vision Tasks in Precision Agriculture. *Comput. Electron. Agric.* **2020**, *178*, 105760. [CrossRef]
225. Gui, J.; Chen, T.; Zhang, J.; Cao, Q.; Sun, Z.; Luo, H.; Tao, D. A Survey on Self-Supervised Learning: Algorithms, Applications, and Future Trends. *arXiv* **2023**, arXiv:2301.05712v3. [CrossRef] [PubMed]
226. Yin, S.; Xi, Y.; Zhang, X.; Sun, C.; Mao, Q. Foundation Models in Agriculture: A Comprehensive Review. *Agriculture* **2025**, *15*, 847. [CrossRef]
227. Khan, A.; Rauf, Z.; Sohail, A.; Khan, A.R.; Asif, H.; Asif, A.; Farooq, U. A Survey of the Vision Transformers and Their CNN–Transformer Based Variants. *Artif. Intell. Rev.* **2023**, *56* (Suppl. 3), 2917–2970. [CrossRef]
228. Chen, J.; Lu, Y.; Yu, Q.; Luo, X.; Adeli, E.; Wang, Y.; Lu, L.; Yuille, A.L.; Zhou, Y. TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. *arXiv* **2021**, arXiv:2102.04306. [CrossRef]
229. Ming, Y.; Meng, X.; Fan, C.; Yu, H. Deep Learning for Monocular Depth Estimation: A Review. *Neurocomputing* **2021**, *438*, 14–33. [CrossRef]

230. Cui, X.-Z.; Feng, Q.; Wang, S.-Z.; Zhang, J.-H. Monocular Depth Estimation with Self-Supervised Learning for Vineyard Unmanned Agricultural Vehicle. *Sensors* **2022**, *22*, 721. [CrossRef]
231. Yang, J.; Guo, X.; Li, Y.; Marinello, F.; Ercisli, S.; Zhang, Z. A Survey of Few-Shot Learning in Smart Agriculture: Developments, Applications, and Challenges. *Plant Methods* **2022**, *18*, 28. [CrossRef]
232. Guldenring, R.; Nalpantidis, L. Self-Supervised Contrastive Learning on Agricultural Images. *Comput. Electron. Agric.* **2021**, *191*, 106510. [CrossRef]
233. Kirillov, A.; He, K.; Girshick, R.; Rother, C.; Dollár, P. Panoptic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Denver, CO, USA, 6 June 2019. [CrossRef]
234. Fu, L.; Gao, F.; Wu, J.; Li, R.; Karkee, M.; Zhang, Q. Application of Consumer RGB-D Cameras for Fruit Detection and Localization in Field: A Critical Review. *Comput. Electron. Agric.* **2020**, *177*, 105687. [CrossRef]
235. Szeliski, R. *Computer Vision: Algorithms and Applications*, 2nd ed.; Springer: Cham, Switzerland, 2022.
236. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the 26th International Conference on Neural Information Processing Systems (NeurIPS), Tahoe, CA, USA, 3–8 December 2012.
237. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 24–27 June 2014; pp. 580–587. [CrossRef]
238. Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3523–3542. [CrossRef]
239. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017. [CrossRef]
240. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016. [CrossRef]
241. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440. [CrossRef]
242. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008. [CrossRef]
243. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2015**, arXiv:1506.01497v3. [CrossRef]
244. Terven, J.; Cordova-Esparza, D. A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond. *arXiv* **2023**, arXiv:2304.00501.
245. Jocher, G.; Ultralytics Team. YOLO by Ultralytics. Available online: https://github.com/ultralytics/ultralytics (accessed on 12 October 2025).
246. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. YOLACT: Real-Time Instance Segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9156–9165. [CrossRef]
247. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention (MICCAI), Munich, Germany, 5–9 October 2015; pp. 234–241. [CrossRef]
248. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S. Transformers in Vision: A Survey. *ACM Comput. Surv.* **2022**, *54*, 200. [CrossRef]
249. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 213–229. [CrossRef]
250. Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; et al. Swin Transformer V2: Scaling Up Capacity and Resolution. *arXiv* **2021**, arXiv:2111.09883.
251. Zhang, M.; Xu, S.; Han, Y.; Li, D.; Yang, S.; Huang, Y. High-Throughput Horticultural Phenomics: The History, Recent Advances and New Prospects. *Comput. Electron. Agric.* **2023**, *213*, 108265. [CrossRef]
252. Huang, Y.; Ren, Z.; Li, D.; Liu, X. Phenotypic Techniques and Applications in Fruit Trees: A Review. *Plant Methods* **2020**, *16*, 107. [CrossRef] [PubMed]
253. Zhang, Z.; Igathinathane, C.; Li, J.; Cen, H.; Lu, Y.; Flores, P. Technology Progress in Mechanical Harvest of Fresh Market Apples. *Comput. Electron. Agric.* **2020**, *175*, 105606. [CrossRef]
254. Mail, M.F.; Maja, J.M.; Marshall, M.; Cutulle, M.; Miller, G.; Barnes, E. Agricultural Harvesting Robot Concept Design and System Components: A Review. *AgriEngineering* **2023**, *5*, 777–800. [CrossRef]
255. Yang, Y.; Han, Y.; Li, S.; Yang, Y.; Zhang, M.; Li, H. Vision-Based Fruit Recognition and Positioning Technology for Harvesting Robots. *Comput. Electron. Agric.* **2023**, *213*, 108258. [CrossRef]

256. Meshram, A.T.; Vanalkar, A.V.; Kalambe, K.B.; Badar, A.M. Pesticide Spraying Robot for Precision Agriculture: A Categorical Literature Review and Future Trends. *J. Field Robot.* **2022**, *39*, 53–171. [CrossRef]
257. Dange, K.M.; Bodile, R.M.; Varma, B.S. A Comprehensive Review on Agriculture-Based Pesticide Spraying Robot. In Proceedings of the International Conference on Sustainable and Innovative Solutions for Current Challenges in Engineering and Technology, Gwalior, India, 20–21 October 2023. [CrossRef]
258. Zahid, A.; Mahmud, M.S.; He, L.; Heinemann, P.; Choi, D.; Schupp, J. Technological Advancements towards Developing a Robotic Pruner for Apple Trees: A Review. *Comput. Electron. Agric.* **2021**, *189*, 106383. [CrossRef]
259. He, L.; Schupp, J. Sensing and Automation in Pruning of Apple Trees: A Review. *Agronomy* **2018**, *8*, 211. [CrossRef]
260. Zeng, H.; Yang, J.; Yang, N.; Huang, J.; Long, H.; Chen, Y. A Review of the Research Progress of Pruning Robots. In Proceedings of the IEEE International Conference on Data Science and Computer Application (ICDSCA), Dalian, China, 28–30 October 2022; pp. 1069–1073. [CrossRef]
261. Koirala, A.; Walsh, K.B.; Wang, Z.; McCarthy, C. Deep Learning—Method Overview and Review of Use for Fruit Detection and Yield Estimation. *Comput. Electron. Agric.* **2019**, *162*, 219–234. [CrossRef]
262. Maheswari, P.; Raja, P.; Apolo-Apolo, O.E.; Pérez-Ruiz, M. Intelligent Fruit Yield Estimation for Orchards Using Deep Learning-Based Semantic Segmentation Techniques—A Review. *Front. Plant Sci.* **2021**, *12*, 684328. [CrossRef]
263. Farjon, G.; Huijun, L.; Edan, Y. Deep Learning-Based Counting Methods, Datasets, and Applications in Agriculture: A Review. *Precis. Agric.* **2023**, *24*, 1683–1711. [CrossRef]
264. Villacrés, J.; Viscaino, M.; Delpiano, J.; Vougioukas, S.; Auat Cheein, F. Apple Orchard Production Estimation Using Deep Learning Strategies: A Comparison of Tracking-by-Detection Algorithms. *Comput. Electron. Agric.* **2023**, *204*, 107513. [CrossRef]
265. Li, X.; Qiu, Q. Autonomous Navigation for Orchard Mobile Robots: A Rough Review. In Proceedings of the Youth Academic Annual Conference of Chinese Association of Automation (YAC), Nanchang, China, 28–30 May 2021; pp. 552–557. [CrossRef]
266. Wang, T.; Chen, B.; Zhang, Z.; Li, H.; Zhang, M. Applications of Machine Vision in Agricultural Robot Navigation: A Review. *Comput. Electron. Agric.* **2022**, *198*, 107085. [CrossRef]
267. Lei, X.; Yuan, Q.; Xyu, T.; Qi, Y.; Zeng, J.; Huang, K.; Sun, Y.; Herbst, A.; Lyu, X. Technologies and Equipment of Mechanized Blossom Thinning in Orchards: A Review. *Agronomy* **2023**, *13*, 2753. [CrossRef]
268. Gené-Mola, J.; Sanz-Cortiella, R.; Rosell-Polo, J.R.; Morros, J.R.; Ruiz-Hidalgo, J.; Vilaplana, V.; Gregorio, E. Fuji-SfM Dataset: A Collection of Annotated Images and Point Clouds for Fuji Apple Detection and Location Using Structure-from-Motion Photogrammetry. *Data Brief* **2020**, *30*, 105591. [CrossRef]
269. Dias, P.A.; Medeiros, H. Semantic Segmentation Refinement by Monte Carlo Region Growing of High Confidence Detections. *arXiv* **2018**, arXiv:1802.07789. [CrossRef]
270. Westling, F. *Avocado Tree Point Clouds Before and After Pruning*; Mendeley Data: Amsterdam, The Netherlands, 2021; Version 1. [CrossRef]
271. Yang, T.; Zhou, S.; Huang, Z.; Xu, A.; Ye, J.; Yin, J. Urban Street Tree Dataset for Image Classification and Instance Segmentation. *Comput. Electron. Agric.* **2023**, *209*, 107852. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.