frontiers | Frontiers in Plant Science

REVIEW
published: 16 May 2022
doi: 10.3389/fpls.2022.868745

Application of Convolutional Neural Network-Based Detection Methods in Fresh Fruit Production: A Comprehensive Review

Chenglin Wang[1,2], Suchun Liu[2], Yawei Wang[2], Juntao Xiong[3], Zhaoguo Zhang[1]*, Bo Zhao[4], Lufeng Luo[5], Guichao Lin[6] and Peng He[7]*

[1] Faculty of Modern Agricultural Engineering, Kunming University of Science and Technology, Kunming, China, [2] School of Intelligent Manufacturing Engineering, Chongqing University of Arts and Sciences, Chongqing, China, [3] College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China, [4] Chinese Academy of Agricultural Mechanization Sciences, Beijing, China, [5] School of Mechatronic Engineering and Automation, Foshan University, Foshan, China, [6] School of Mechanical and Electrical Engineering, Zhongkai University of Agriculture and Engineering, Guangzhou, China, [7] School of Electronic and Information Engineering, Taizhou University, Taizhou, China

As one of the representative algorithms of deep learning, a convolutional neural network (CNN) with the advantage of local perception and parameter sharing has been rapidly developed. CNN-based detection technology has been widely used in computer vision, natural language processing, and other fields. Fresh fruit production is an important socioeconomic activity, where CNN-based deep learning detection technology has been successfully applied to its important links. To the best of our knowledge, this review is the first on the whole production process of fresh fruit. We first introduced the network architecture and implementation principle of CNN and described the training process of a CNN-based deep learning model in detail. A large number of articles were investigated, which have made breakthroughs in response to challenges using CNN-based deep learning detection technology in important links of fresh fruit production including fruit flower detection, fruit detection, fruit harvesting, and fruit grading. Object detection based on CNN deep learning was elaborated from data acquisition to model training, and different detection methods based on CNN deep learning were compared in each link of the fresh fruit production. The investigation results of this review show that improved CNN deep learning models can give full play to detection potential by combining with the characteristics of each link of fruit production. The investigation results also imply that CNN-based detection may penetrate the challenges created by environmental issues, new area exploration, and multiple task execution of fresh fruit production in the future.

Keywords: computer vision, deep learning, convolutional neural network, fruit detection, fruit production

OPEN ACCESS

Edited by:
Gregorio Egea,
University of Seville, Spain

Reviewed by:
Orly Enrique Apolo-Apolo,
University of Seville, Spain
Mohsen Yoosefzadeh Najafabadi,
University of Guelph, Canada

****Correspondence:***
Juntao Xiong
xiongjt2340@163.com
Zhaoguo Zhang
zhaoguozhang@163.com

Specialty section:
This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science

Received: *03 February 2022*
Accepted: *03 March 2022*
Published: *16 May 2022*

Citation:
Wang C, Liu S, Wang Y, Xiong J, Zhang Z, Zhao B, Luo L, Lin G and He P (2022) Application of Convolutional Neural Network-Based Detection Methods in Fresh Fruit Production: A Comprehensive Review. Front. Plant Sci. 13:868745. doi: 10.3389/fpls.2022.868745

INTRODUCTION

Fresh fruits in the market are beloved by people because of their enticing aroma and unique flavor. From fruit flowers blooming to fruit grading, every link of fresh fruit production needs to be seriously supervised so that fruits enter the market without economic loss. In recent years, the world agricultural population and labor force have been having a declining

trend leading to the urgent need for automation of fresh fruit production (Yuan et al., 2017). Object detection based on computer vision has been applied to the main link of automatic fresh fruit production such as smart yield prediction, automatic harvesting robots, and intelligent fruit quality grading (Naranjo-Torres et al., 2020).

A function of ML is to ensure that machines can automatically detect objects accurately. Although ML has been applied in many fields, the ML technology has been developing to achieve efficient detection. The detection performance of traditional ML will not improve with increase in training sample data. The features need to be given artificially for object detection, which is also a disadvantage of traditional ML (Mohsen et al., 2021). As an intelligent algorithm in the development of ML, DL has significant advantages over traditional algorithms of ML. The detection performance of DL usually improves with increase in the amount of training sample data. DL can automatically extract features of a detected object using network structure. However, DL takes a lot of training time and runs on computers with higher cost configurations compared with traditional ML (Joe et al., 2022).

Deep learning is a further study on artificial neural networks such as deep belief network (Hinton et al., 2006), recurrent neural network (Schuster and Paliwal, 1997), and convolutional neural network (LeCun et al., 1989). The deep learning algorithm has a similar calculation principle with a mechanism of the visual cortex of animals (Rehman et al., 2019). The deep learning-based technology has broad applications in many domains due to its superior performance in operation speed and accuracy, for example, in the medical field (Gupta et al., 2019; Zhao Q. et al., 2019), in the aerospace field (Dong Y. et al., 2021), in the transportation sector (Nguyen et al., 2018), in the agriculture field (Kamilaris and Prenafeta-Boldú, 2018), and in the biochemistry field (Angermueller et al., 2016).

A CNN with a convolutional layer and a pooling layer was proposed by Fukushima (1980), which was subsequently improved to LeNet (LeCun et al., 1998), GoogleNet (Szegedy et al., 2015), ResNet (He et al., 2016), AlexNet (Krizhevsky et al., 2017), and so on. With the appearance of R-CNN (Girshick et al., 2014), CNN-based object detection became a hot research topic on computer vision and digital image processing (Zhao Z. et al., 2019). Object detection is the coalition of object classification and object location requiring a network to differentiate an object region from the background and accomplish the classification and location of the object. The technique of CNN-based image segmentation using a CNN model to perceive the representative object of each pixel for classifying and locating objects can be performed for object detection tasks. Frequently used image segmentation models are Mask-R-CNN, U-Net (Ronneberger et al., 2015), SegNet (Badrinarayanan et al., 2017), DeepLab (Chen et al., 2018), and so on.

Early fruit image segmentation algorithms use traditional ML algorithms to identify fruit objects by combining shallow characteristics of fruits such as color, texture, and shape, and mainly included threshold segmentation (Pal and Pal, 1993), DTI (Quinlan, 1986), SVM (Cortes and Vapnik, 1995), cluster analysis (Tsai and Chiu, 2008), and so on. Color traits of fruits are frequently used in fruit detection (Thendral et al., 2014; Zhao et al., 2016). Shape, as an outstanding mark of fruits, is applied to fruit segmentation and recognition (Nyarko et al., 2018; Tan et al., 2018). In addition, spectral features and depth information are applied in fruit detection (Bulanon et al., 2009; Okamoto and Lee, 2009; Gené-Mola et al., 2019a; Lin et al., 2019; Tsoulias et al., 2020). The above methods can detect fruit objects; however, they have certain limitations of features expression for fruit object detection in a complex environment. CNN-based detection technology has been proved to have a potential in fresh fruit production by many studies (Koirala et al., 2019b). Models combined with CNN, for example, CNN + SVM (Dias et al., 2018), CNN + ms-MLP (Bargoti and Underwood, 2017), fuzzing mask R-CNN (Huang et al., 2020), faster R-CNN (Gao et al., 2020), the Alex-FCN model (Wang et al., 2018), and 3D-CNN (Wang et al., 2020), have obtained satisfactory detection results in fruit flower detection, fruit recognition, fruit maturity prediction, and surface defect detection-based fruit grading. These successful studies imply that CNN-based methods can break the technical bottleneck in detection and accelerate the mechanization of fresh fruit production.

As shown in **Figure 1**, this review investigates the CNN-based detection application in the process of fresh fruit production, which is a complete process from fruit flower detection, growing fruit detection, fruit picking to fruit grading. We provide a comprehensive introduction and analysis of the CNN model and its improved models in fresh fruit production. In addition, different CNN-based detection methods are compared and summarized in each link of fresh fruit production. The arrangement of this article is as follows: Section "Common Models and Algorithms of Convolutional Neural Network" introduces the composition and algorithms of CNN; Section "Implementation Process of Convolutional Neural Network-Based Detection" explains the CNN-based detection implementation process; Section "Convolutional Neural Network-Based Fresh Fruit Detection" investigates the current research on CNN applications in each link of fresh fruit production; Section "Challenges and Future Perspective" discusses difficulties that will be encountered by CNN-based detection in future research on fresh fruit

Abbreviations: CNN, convolutional neural network; DBN, deep belief network; RNN, recurrent neural network; VGG, visual geometry group; DTI, decision tree induction; SVM, support vector machine; 2D, two-dimensional; ms-MLP, multiscale-multilayered perceptron; HoG, histogram of oriented gradient; ML, machine learning; GLCM, gray-level co-occurrence matrix; CIELab, Commission Internationale de l'Eclairage Laboratory; CHT, circular Hough transform; SLIC, simple linear iterative clustering; YOLO, you only look once; SSD, single shot multibox detector; mAP, mean average precision; STN, Special Transform Network; CCD, charge coupled device; SMOTE, synthetic minority oversampling technique; DC-GAN, deep convolutional generative adversarial network; CycleGAN, cycle generative adversarial network; CVAE-GAN, conditional autoencoder generative adversarial network; GAN, generative adversarial network; CPU, central processing unit; GPU, graphics processing unit; TP, true positive; FN, false negative; FP, false positive; TN, true negative; MAE, mean absolute error; MSE, mean square error; RMSE, root mean square error; FCN, full convolutional net; AV, unmanned aerial vehicle; MS-FRCNN, multiple scale Faster R-CNN; MIoU, mean intersection over union; SFM, structure from motion; ROI, region of interest; E-CNN, ensemble-convolutional neural net; NIR, near infrared.

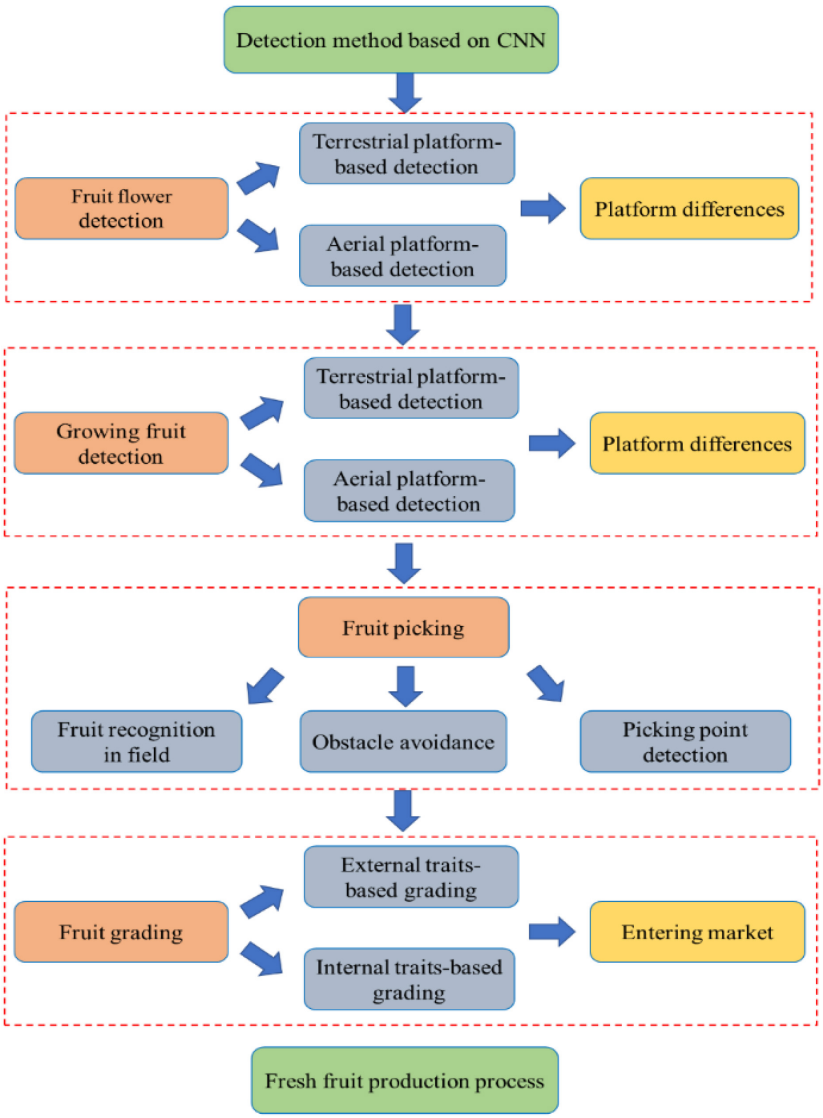

FIGURE 1 | Convolutional neural network (CNN)-based detection application in main links of fresh fruit production.

production; Section "Conclusion" presents an entire summary of this investigation.

COMMON MODELS AND ALGORITHMS OF CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Network Models for Image Detection

Common CNN models used for image detection are usually composed of convolutional layers, activation functions, pooling layers, and full-connected layers (Mohsen et al., 2021). A CNN model transforms an image into high dimension information, so a computer can read and extract features from the image. In two-dimensional (2D) convolution operation, each pixel value of an input image entering into a convolutional layer is convoluted with a kernel to generate a feature map. When an input image is three-dimensional (3D) or four-dimensional (4D), a multi-dimension convolution operation will be implemented. In the multi-dimension convolution operation, the channel number of kernels is equal to the channel number of input images, and the channel number of output feature maps is the number of kernels (Alzubaidi et al., 2021). However, in convolutional layers and full-connected layers, the linear connection between the input and the output restricts the ability of a CNN model to solve more complex problems. The activation function is added after the operations of convolution layers and full-connected layers, which can capacitate a CNN model to solve non-linear problems. Common activation functions include the Sigmoid function, the Tanh function, the ReLU function, SoftMax, and so on.

TABLE 1 | Structure and performance of common convolutional neural network (CNN) models for image detection.

CNN models	Weight layers	Convolutionlayer	Kernel size	Active function	Dropout[a]	LRN[b]	BN[c]	Top-5 error (on ImageNet)
AlexNet	8	5	3×3, 5×5, 11×11	ReLU	√	√	–	16.4%
VGG	19	16	3×3	ReLU	√	–	–	7.3%
GoogleNet (Inception-V1)	22	21	1×1, 3×3, 5×5, 7×7	ReLU	√	√	–	6.7%
ResNet	152	151	1×1, 3×3, 7×7	ReLU	√	–	√	3.57%
DenseNet	265	264	1×1, 3×3, 7×7	ReLU	√	–	√	5.29%
MobileNet	28	27	1×1, 3×3	ReLU	–	–	√	*

[a]*Dropout is a training trick, which means that neural network units are temporarily discarded from the network according to a certain probability in the training process of a deep learning network.*
[b]*LRN, local response normalization, is a training trick that can enhance the generalization ability of a model. It creates a competitive mechanism for activities of local neurons, which can make the value of neurons with large responses larger and inhibit neurons with small feedback.*
[c]*BN, batch normalization, normalizes the data of each layer and performs linear transformation to improve data distribution.*
**Means that we have not found relevant data about Mobilenet in the public references.*

LeNet is the first improved CNN; however, it has not been widely promoted and applied because of simple network structure (LeCun and Bengio, 1995). AlexNet is the first deep CNN architecture and the first CNN model trained on GPU (Krizhevsky et al., 2017). A VGG model with four network structures and different configurations was proposed by the Visual Geometry Group of Oxford University in 2014 (Simonyan and Zisserman, 2014). The most popular network among VGG models is VGG-16 containing thirteen convolutional layers and three full-connected layers. GoogLeNet was a new deep learning structure proposed in 2014 (Szegedy et al., 2015). The most unique of GoogLeNet is the inception component, which utilizes partial connection to accomplish parameter reduction and computation simplicity. A series of inception components including InceptionV2, InceptionV3, and InceptionV4, was proposed for optimizing GoogLeNet (Szegedy et al., 2016). By proving the existence of degradation of CNN while its depth is increasing, ResNet was proposed to improve the CNN by designing residual components with the shortcut connection (He et al., 2016). DenseNet was proposed in 2017, and dense block was the highlight of DenseNet by building connections of all layers with each other to ensure maximum information flow among the layers (Huang et al., 2017). With the popularization of CNN models, it is required that CNN-based image recognition tasks are implemented on mobile terminals or embedded devices. As a lightweight model, MobileNet was designed to run on the CPU platform, and it had good detection accuracy (Howard et al., 2017). These models are fundamentals of CNN-based object detection and can help computers learn more information about images because of functions of feature recognition and extraction. The structure and image detection performance of the above common CNN models are summarized in **Table 1**.

Convolutional Neural Network Models for Three-Dimensional Point Cloud Detection

With the development of vision technology, sensors that directly acquire 3D data are becoming more common in robotics, autonomous driving, and virtual/augmented reality applications. Because depth information can eliminate a lot of segmentation ambiguities in 2D images and provides important geometric information, the ability to directly process 3D data is invaluable in these applications. However, 3D data often come in the form of point clouds. Point clouds are typically represented by a set of 3D points that are not arranged in order, each with or without additional features (such as RGB color information). Because of the disordered nature of point clouds and the fact that they are arranged differently from regular mesh-like pixels in 2D images, traditional CNNs struggle to handle this disordered input.

At present, the deep learning point cloud target recognition method mainly has three kinds of point cloud target recognition methods based on views (Kalogerakis et al., 2017), voxels (Riegler et al., 2016), and point clouds (Qi et al., 2017a). Among them, the idea based on views is still to convert three-dimensional data into a two-dimensional representation; that is, 3D data are projected according to different coordinates and different perspectives to obtain a two-dimensional view, and then the two-dimensional image convolution processing method is used to extract features from each view and, finally, aggregate the features to obtain classification and segmentation results. The idea based on voxels is to put an unordered point cloud into the voxel grid, so that it becomes a three-dimensional grid regular data structure, and then as network input data. However, in order to solve problems of view-based and voxel-based computational complexity and information loss, researchers began to consider directly inputting raw point cloud data into the network for processing.

At Stanford University in the United States, Qi et al. (2017a) proposed a new type of neural network, PointNet, for point cloud identification and segmentation directly using a point cloud as the input object, the spatial transformation network T-Net to ensure the displacement invariance of the input point, a shared multilayer perceptron (MLP) to learn the characteristics of each point, and, finally, the maximum pooling layer to aggregate global features. However, PointNet cannot learn the relationship characteristics between different points in the local neighborhood, and then Qi et al. (2017b) proposed PointNet++ to improve PointNet, according to the idea of two-dimensional convolution proposed hierarchical point cloud feature learning for local areas, which is composed of sampling layer, grouping layer and feature extraction layer (PointNet) in the hierarchical module, while improving the stability of the network architecture

and the ability to obtain details. Later, the description ability of local features was enhanced in order to make the local structure information between points, such as distance and direction, be able to learn in the network.

PointNet inputs an irregular point cloud directly into the deep convolutional network, the framework represents the point cloud as a set of 3D points $\{(P|i = 1, \ldots, n\}$, where each point P is its 3D coordinates plus additional feature channels such as color, normal vector, and other information; the architecture is shown in **Figure 2**. In response to the point cloud disorder problem, PointNet pointed out that a symmetric method is used; that is, maximum pooling, no matter how many orders there are in N points, the maximum eigenvalue in the pooling window corresponding to N points is selected for each dimension of the final high-latitude feature and fused into the global feature. For the rotation invariance problem of point cloud, PointNet points out that spacial transform network (STN) is used to solve it. Through the T-Net network to learn the point cloud itself attitude information to obtain a DD rotation matrix (D represents the characteristic dimension), PointNet in the input space transformation using 3×3, feature space transformation using 64×64 to achieve the most effective transformation for the target.

Convolutional Neural Network-Based Detection Algorithms

Convolutional neural network-based detection algorithms mainly include object detection algorithms, semantic segmentation algorithms, and instance segmentation algorithms, which are described in detail as follows.

Object Detection Algorithms

As a kind of object detection algorithm, a two-stage detector is mainly composed of a region proposal generator and classes and bounding box prediction. The R-CNN series is the most representative two-stage detector and includes R-CNN (Girshick et al., 2014), Fast-R-CNN (Girshick, 2015), Faster-R-CNN (Ren et al., 2017), etc. R-CNN is the pioneer in using deep learning for object detection. After that, researchers proposed Fast-R-CNN and Faster-R-CNN in succession to update detection performance. **Figure 3** shows the structure of Faster-R-CNN, which is frequently used. Besides the above object detection algorithms, R-FCN and Libra R-CNN are also two-stage detectors.

Compared with a two-stage detector, a one-stage detector conducts classification and bounding box regression after feature extraction without generation of proposal regions. Prediction of objects depends on doing dense sampling on an input picture. Representative one-stage detectors are the YOLO series and SSD (single shot multibox detector). The YOLO series contains YOLOv1 (Redmon et al., 2016), YOLOv2 (Redmon and Farhadi, 2017), YOLOv3 (Redmon and Farhadi, 2018), and YOLOv4 (Bochkovskiy et al., 2020). Notably, during the evolution of YOLO, a new convolution neural net, DarkNet, was constructed for feature extraction. Furthermore, YOLOv2 referenced the anchor conception from Faster-R-CNN. YOLOv3 contains three different output nets that can predict multi-scale pictures. SSD (Liu W. et al., 2016) is also a kind of one-stage detector that can implement multi-box prediction. VGG-16 was used as a backbone in SSD. With the development of DL, more improved one-stage detection algorithms have been designed.

A comparison of CNN models between two-stage detectors and one-stage detectors is shown in **Table 2**. As can be seen in **Table 2**, frames per second (FPS) of the one-stage detector are bigger than those of the two-stage detector, which implies that the detection speed of the one-stage detector is faster than that of the two-stage detector. The FPS and mAP of the Mask-R-CNN model are bigger than those of other models of the two-stage detector. It shows that the Mask-R-CNN model has faster detection speed and higher detection accuracy than the two-stage detector. However, in the one-stage detector, no CNN model has faster detection speed and higher detection accuracy. Because of lack of mAP in some CNN models on data of VOC2012 and COCO, the accuracy of the two detectors cannot be compared.

Semantic Segmentation Algorithms

Unlike box recognition in object detection, semantic segmentation refers to pixel-level recognition and classification, which classifies pixels of the same class into one group. Early DL-based semantics segmentation methods performed clustering to generate super-pixels and a classifier to classify them (Couprie et al., 2013; Farabet et al., 2013). However, such methods have drawbacks of time-consuming and rough segmentation results. With the popularity and development of object detection algorithms based on CNNs, semantic segmentation algorithms have also made great progress, and can be divided into region-classification-based image semantic segmentation and pixel-classification-based image semantic segmentation.

The method of region-classification-based image semantic segmentation first selects the appropriate region, then classifies the pixels in the candidate region. SDS (simultaneous detection and segmentation) is a model based on R-CNN that can

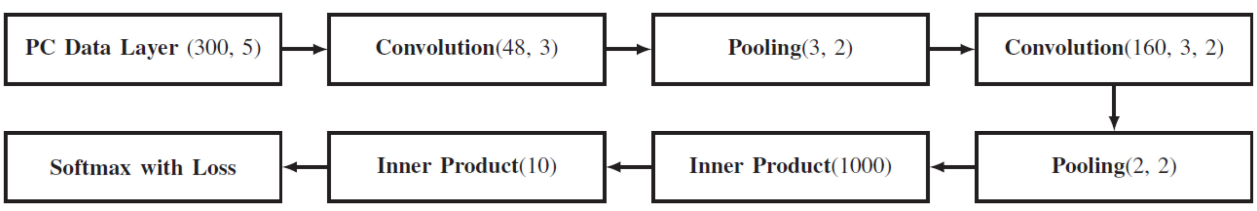

FIGURE 2 | Structure of PointNet.

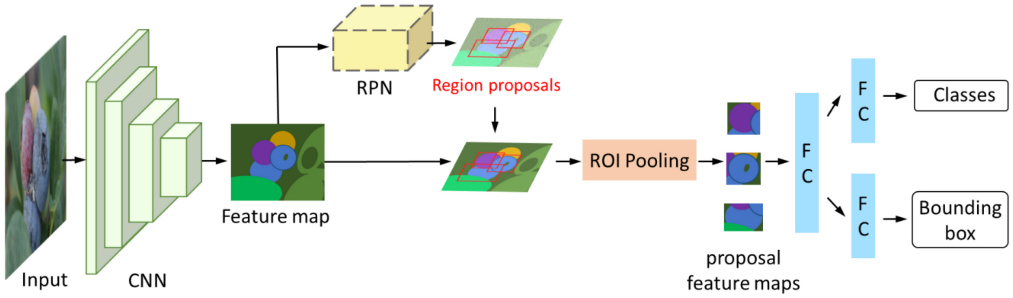

FIGURE 3 | Faster-R-CNN structure. The feature map is extracted by a convolutional neural network, and then the RPN (region proposal network) generates several accurate region proposals according to the feature map. The region proposals are mapped to the feature map. The ROI (region of interest) pooling layer is responsible for collecting proposal boxes and calculating proposal feature maps. Finally, the category of each proposal is predicted through the FC (full connect) layer.

TABLE 2 | Summary of common CNN-based object detection models.

Type	Name	Backbone	Bounding boxes generation	Additional blocks	FPS[a]	mAP/%		References
						VOC2012[b]	COCO[c]	
Two-stage	R-CNN	AlexNet	SS[d]	–	0.03	59.2	–	Girshick et al., 2014
	Fast-R-CNN	VGG-16	SS+ROI pooling	–	7.00	68.4	19.7	Girshick, 2015
	Faster-R-CNN	VGG-16/ResNet-101	RPN+ROI pooling	–	7.00/5.00	70.4/73.8	21.9/34.9	Ren et al., 2017
	Mask-R-CNN	ResNeXt-101-FPN	RPN+ROI align	FCN	11.00	73.9	39.8	He K. et al., 2017
One-stage	SSD	VGG-16	Anchor	–	19.3	78.5	28.8	Liu W. et al., 2016
	YOLOv1	GoogleNet	–	–	45.0	57.9	–	Redmon et al., 2016
	YOLOv2	DarkNet-19	Anchor	–	40.0	73.5	21.6	Redmon and Farhadi, 2017
	YOLOv3	DarkNet-53	Anchor	FPN, SPP	51.0	–	33.0	Redmon and Farhadi, 2018
	YOLOv4	CSPDarkNet53	Anchor	FPN+PA, SPP	23.0	–	43.5	Bochkovskiy et al., 2020

[a] *FPS, frames per second, is used to measure how many frames (pictures) the target network can detect per second.*
[b] *VOC2012: a dataset used in pattern analysis, statistical modeling, and computational learning visual object classes challenge 2012.*
[c] *COCO: Microsoft Common Objects in Context, a dataset funded and labeled by Microsoft in 2014.*
[d] *SS: selective search (Uijlings et al., 2013).*

simultaneously detect and semantically segment targets (Hariharan et al., 2014). In 2016, based on the SDS method, Liu S. et al. (2016) convoluted images using sliding windows of different sizes and constructed multi-scale feature maps, proposed an MPA (multi-scale patch aggregation) method that can semantically segment an image at the instance level. DeepMask is a segmentation model proposed based on CNN to generate object proposals (Pinheiro et al., 2015). It generates image patches directly from original image data and then generates a segmentation mask for given image patches. The whole process is applied to a complete image to improve the efficiency of segmentation.

The method of pixel-classification-based semantic segmentation does not need to generate object candidate regions but extracts image features and information from labeled images. Based on that information, a segmentation model can learn and infer the classes of pixels in an original image, and classify each pixel in the image directly to achieve end-to-end semantic segmentation. FCN (fully convolutional network) is a popular semantic segmentation model that can be compatible with any size of images (Shelhamer et al., 2017). FCN can distinguish the categories of pixels directly, which greatly promotes the development of semantic segmentation. Subsequently, researchers proposed a series of methods based on FCN. FCN-based image semantic segmentation methods are as follows: DeepLab, DeepLab-V2, and DeepLab-V3. Image semantics segmentation methods based on encoder-decoder model are as follows: U-net, Segnet, Deconvnet, and GCN (global convolution network).

Instance Segmentation Algorithms

The purpose of instance segmentation is to distinguish different kinds of objects in an image and different instances of the same kind. Therefore, it has the characteristics of object detection and semantic segmentation at the same time. Because of the characteristics of instance segmentation, it can include instance segmentation based on object detection and instance segmentation based on semantics segmentation.

An instance segmentation algorithm based on object detection has been the mainstream direction in the field of instance segmentation research in recent years. Its main process is to locate an instance using an object detection algorithm, and then segment the instance in each detected box. Mask-R-CNN is one of the famous models in instance segmentation proposed by He K. et al. (2017). Mask-R-CNN is one of the famous models in instance segmentation on the basis of Fast-R-CNN(He K. et al.,

2017). As a representative instance segmentation model, many scholars are deeply inspired by Mask-R-CNN. Based on Mask-R-CNN, PANet (path aggregation network) introduces a bottom-up path augmentation structure, adaptive feature pooling, and a fully connected fusion structure to obtain more accurate segmentation results (Liu S. et al., 2018). Chen et al. (2018) proposed Masklab, which uses directional features to segment instances of the same semantic class. In 2019, the first instance segmentation algorithm based on a one-stage object detection algorithm, YOLACT, was proposed by Bolya et al. (2019). It added a mask generation branch behind the one-stage object detector to complete a segmentation task. The overall structure of YOLACT is relatively lightweight, and the trade-off between speed and effect would be good. In addition, there are some newly proposed instance segmentation algorithms such as MS-R-CNN (Huang et al., 2019), BMask-R-CNN (Cheng et al., 2020) and BPR (Tang et al., 2021).

An instance segmentation algorithm based on semantic segmentation classifies each pixel first and then segments different instances of the same category. For example, the SGN (Liu et al., 2017) model decomposes an instance segmentation into multiple subtasks, then uses a series of neural networks to complete these subtasks, and finally recombines the results of the subtasks to obtain the segmentation task.

Differences of Detection Algorithms

In this section, differences among object detection, semantic segmentation, and instance segmentation are visually explained through pear flower detection. **Figure 4A** is an undetected image of pear flowers. The result of detecting pear flowers with the object detection algorithm is shown in **Figure 4B**, and it shows the approximate position of pear flowers with bounding boxes. The result with semantic segmentation algorithm is shown in **Figure 4C**, which reaches the pixel level compared with the result of object detection. It means that when labeling data sets, the annotation of the task of semantic segmentation is also at pixel level. Compared with rectangular box annotation in the object detection task, the annotation of semantic segmentation task is more complex. The result with the instance segmentation algorithm is shown in **Figure 4D**, and the detection results of instance segmentation are more detailed than those of semantic segmentation in distinguishing each pear flower individual.

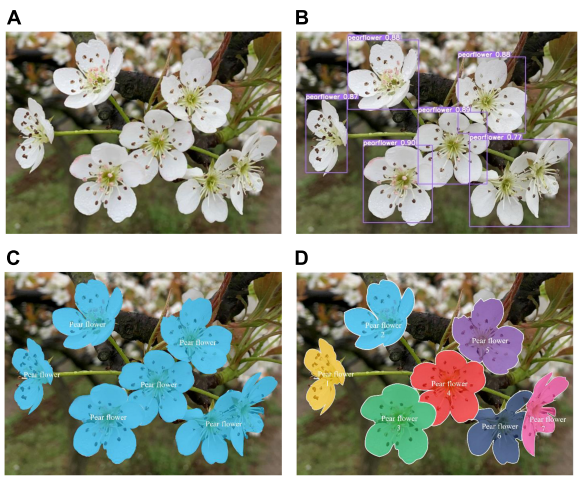

FIGURE 4 | Different CNN-based algorithms for pear flower detection. **(A)** Original image, **(B)** object detection, **(C)** semantic segmentation, and **(D)** instance segmentation.

IMPLEMENTATION PROCESS OF CONVOLUTIONAL NEURAL NETWORK-BASED DETECTION

This section introduces the main procedures of comprehensively training a CNN-based deep learning model for basic tasks. The first step is determining the learning target and establishing the data set. Second, it is vital to choose an adept deep learning framework to modify the model and implement training. Finally, mastering the estimation metrics of deep learning models leads to knowing the performance of the modified models and training results.

Data Set Construction

Dataset Acquisition

An RGB camera, which can capture the properties of a fruit surface, such as color, shape, defect, and texture, is a pervasive and affordable camera for image acquisition used in many types of research (Fu et al., 2020a). Vasconez et al. (2020) held an RGB camera and acquired apple, avocado, and lemon pictures at 30 frames per second in orchards. However, the information obtained from RGB images is not sufficient for 3D location and reconstruction. Thus, most researchers have begun utilizing RGB-D to capture RGB images and depth images in their experiments. RGB-D cameras generally operate with three depth measurement principles: structured light, time of flight, and active infrared stereo technique (Fu et al., 2020a). Data sets that provide geometric information and radiation information can enhance the models' ability to distinguish fruits from complex environments. Gené-Mola et al. (2019b) established an apple data set containing multimodal RGB-D images and pointed out that the model provided with RGB-D images is more robust than that provided with RGB images in a complex environment. However, sensors in most depth cameras cannot obtain information beyond 3.5 m, and light detection and ranging (LiDAR) scanners are needed to acquire information at a far distance (Tsoulias et al., 2020). A LiDAR scanner can directly provide three-dimensional positioning information of fruits without being affected by light conditions. In addition, LiDAR data can improve the positioning accuracy of fruits because of the appearance of different objects showing different reflectivity to laser. Gené-Mola et al. (2019a), by detecting Fuji apples in orchards with LiDAR, found that the reflection of apple surface was 0.8 higher than that of leaves and branches at a wavelength of 905 nm.

The internal properties of fruits need hyperspectral reflectance images to be represented. Yu et al. (2018) used a hyperspectral imaging system that constituted of a spectrometer, a CDD camera, a light system, and a computer to detect the internal

features of Korla fragrant pear. Some scholars bought a designed hyperspectral system for data collection (Wang et al., 2020).

Data Set Augmentation

Data sets, as an input, play a significant part in a DL model. Most researchers consider that enhancing the scale and quality of data sets can strengthen the models' generalization and learning capacity. The methods of dataset augmentation can be divided into the basic-image-manipulation-based method and the DL-based method. The most straightforward and frequently-used methods based on basic image processing are geometric transformations, flipping, color space, cropping, rotation, translation, noise injection, color space transformations, kernel filters, mix images, and random erasing. **Figure 5** displays example images with some usual image processes. In addition, the DL-based method contains SMOTE (Chawla et al., 2002), adversarial training, DC-GAN (deep convolutional GAN) (Zheng et al., 2017), CycleGAN (Zhu et al., 2017), CVAE-GAN (Bao et al., 2017), etc.

Some researchers processed images from angle, brightness, and sharpness to simulate different light conditions (Jia et al., 2020). Some used clockwise rotation, horizontal mirror, color balance processing, and blur processing to augment a data set for apple detection (Tian et al., 2019). Flowers have distinct characteristics from fruit organs. Thus, Tian et al. (2020) proposed a novel image augmentation method as per apple inflorescence (**Figure 6**). The procedure of image generation is displayed in **Figure 7**. They clipped 50 pictures of central flowers and 150 pictures of side flowers. Then, they filtered and combined these clipped images to generate foreground pictures. At the same time, 200 pictures were extracted and processed for background pictures. Finally, sample images were produced by coalescing foreground pictures and background pictures. The experiment results proved that this way of augmentation contributed to detection performance.

Convolutional Neural Network Model Training

Training Tools

It is onerous to construct a deep learning model from zero. Many open-source or commercial deep learning tools came into being with the advent of deep learning (Li et al., 2021). In the field of fresh fruit detection, Caffe, TensorFlow, Keras, and PyTorch are popular open-source training tools.

Caffe is the abbreviation of convolution architecture for feature extraction, and is one of the earlier DL frameworks. Caffe defines a network structure in the form of configuration text

FIGURE 5 | Example images with different image processes. **(A)** Original image, **(B)** vertical flip image, **(C)** noise injected image, **(D)** sharpened image, **(E)** Gaussian blurry image, **(F)** random erased image, **(G)** image with brightness adjustment, **(H)** RGB2GRB image, and **(I)** gray image.

FIGURE 6 | Apple inflorescence: **(A)** the central flower and the side flowers have a bud shape, **(B)** the central flower has a semi-open shape and the side flowers have a bud shape, **(C)** the central flower has a fully open shape and the side flowers have bud and semi-open shapes, and **(D)** the central flower and the side flowers have a fully open shape.

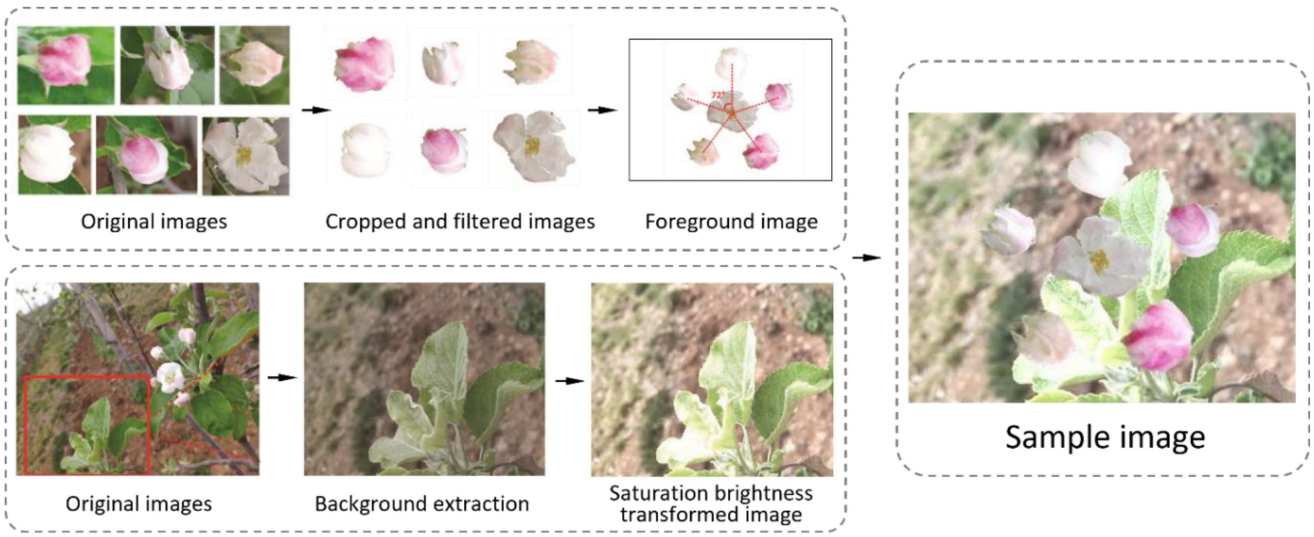

FIGURE 7 | Procedure of image generation in Tian et al. (2020).

instead of code. Users can expand new models and learning tasks with its modular components (Jia et al., 2014). TensorFlow is an open-source machine learning library from Google Brain that can be used for a variety of deep learning tasks, including CNN, RNN, and GAN (generative adversarial network) (Abadi et al., 2016). It uses data flow graphs to represent calculations, shared states, and operations (Zhu et al., 2018). Keras is a very friendly and simple DL framework for beginners. Strictly speaking, it is not an open-source framework but a highly modular neural network library based on TensorFlow and Theano. PyTorch is a DL framework launched by Facebook in 2017 and is based on the original Torch framework; it utilizes Python as main development language (Paszke et al., 2019). Furthermore, the open-source code of Caffe2 has merged into PyTorch, which signifies that PyTorch has strong capacity and flexibility. **Table 3** describes the detail and differences of the above DL tools. In **Table 4**, we display the code of the first convolutional layer of Lenet-5 in different languages.

Furthermore, data set annotation, which generates ground truth for supervising networks' learning object features, is a prerequisite for tasks of object detection and segmentation. Familiar label tools have LabelImg, LabelMe (Russell et al., 2007), Matlab, Yolo_mark, Vatic, CVAT, etc.

Parameter Tuning

Parameter initialization is very important. Reasonable initial parameters can help a model improve training speed and avoid local minima. The Kaiming initialization and Glorot initialization methods are generally used (Glorot and Bengio, 2010; He et al., 2015).

In the beginning of the training, all parameters have typically random values and, therefore, far away from the final solution. Using a too-large learning rate may result in numerical instability. We can use warm-up heuristic (He et al., 2019) to gradually increase the learning rate parameter from 0 to the initial learning rate, and then use the conventional learning rate attenuation scheme. With the progress of training, a model will gradually converge to the global optimum. It is necessary to reduce the learning rate to prevent a model from oscillating back and

TABLE 3 | Comparison of Caffe, TensorFlow, Keras, and PyTorch.

Name	Caffe	TensorFlow	Keras	PyTorch
Support language	C++/Python/MATLAB	C++/Python	Python	Python
Support hardware	CPU/GPU	CPU/GPU/Mobile	CPU/GPU/Mobile	CPU/GPU
Support system	Linux/Windows/MacOS	Linux/Windows/MacOS/Android/IOS	Linux/Windows/MacOS/Android/IOS	Linux/Windows/MacOS
Traits	Strong readability and expansibility, stable and superior performance	Comprehensive functionality, good visualization, and active user community	Highly modular, keeping each module short and simple, and ease of extension.	Intuitive design, ease of use, and active user community

forth near the optimum. Generally, learning rate adjustment strategies such as Step, MultiStep, and exponential and cosine annealing can be used.

Selection of an optimizer plays an important role in DL training and is related to whether the training can converge quickly and achieve high accuracy and recall. Commonly used optimizers include gradient descent, momentum, SGD, SGDM, Adagrad, Rmsprop, Adam, etc.

Convolutional neural network learning needs to establish millions of parameters and a large number of labeled images. If the amount of data is not enough, a model will be over fitted, and the effect is likely to be worse than traditional manual features. If the data set of a new task is significantly different from the original data set and the amount of data is small, one can try transfer learning to complete the new task (Oquab et al., 2014). The weight update of a whole network can be adopted during transfer learning.

TABLE 4 | Different languages define the code of the first convolution layer of Lenet-5.

Caffe

```
layer{
    name:"conv1"
    type:"Convolution"
    bottom:"data"
    top:"conv1"
    param{
        lr_mult:1
    }
    param{
        lr_mult:2
    }
    convolution_param{
        num_output:20
        kernel_size:5
        stride:1
        weight_filler{
            type:"xavier"
        }
        bias_filler{
            type:"constant"
        }
    }
}
```

TensorFlow

```
def hidden_layer(input_tensor,regularizer,avg_class,resuse):
    with tf.variable_scope("C1-conv",reuse=resuse):
        conv1_weights=tf.get_variable("weight", [5, 5, 1, 32],
            initializer=tf.truncated_normal_initializer(stddev=0.1))
        conv1_biases=tf.get_variable("bias", [32],
            initializer=tf.constant_initializer(0.0))
        conv1=tf.nn.conv2d(input_tensor, conv1_weights,
        strides=[1, 1, 1, 1],
            padding="SAME")
        relu1=tf.nn.relu(tf.nn.bias_add(conv1, conv1_biases))
```

Keras

```
model.add(Conv2D(filters=6,
        kernel_size=5,
        strides=1,
        activation='relu',
        input_shape=(32,32,1)))
```

PyTorch

```
self.conv2=nn.Sequential(
        nn.Conv2d(in_channels=6, out_channels=16,
        kernel_size=5, stride=1),
        nn.MaxPool2d(kernel_size=2)
    )
```

Evaluation Metrics

The confusion matrix is a basic, intuitive, computational, and simple method for measuring the accuracy of a model. Take the binary classification model as an example, and its confusion matrix is shown in **Figure 8**. It is mainly composed of four basic indicators: TP (true positive), FN (false negative), FP (false positive), and TN (true negative).

- TP: an outcome where a model correctly predicts a positive class.
- FP: an outcome where a model incorrectly predicts a positive class.
- TN: an outcome where a model correctly predicts a negative class.
- FN: an outcome where a model incorrectly predicts a negative class.

With a confusion matrix, accuracy, precision, recall, and F1-score can be calculated to evaluate a model. Accuracy (Eq. 1) indicates the proportion of correctly classified test instances to the total number of test instances. Precision (Eq. 2) represents

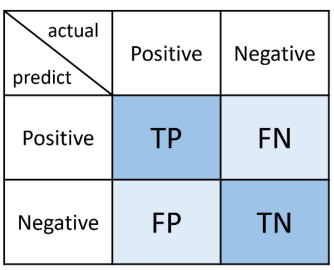

FIGURE 8 | Basic confusion matrix.

the correct proportion of positive samples predicted by a model. Recall (Eq. 3) represents the proportion of all positive samples that are correctly predicted by a model. Generally speaking, precision and recall is a pair of contradictory indicators. As the weighted harmonic average of the two of them, F1-score (Eq. 4) balances the relative importance between precision and recall.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{4}$$

In addition to the above basic evaluation metrics, there are also IoU (intersection over union) and mAP (mean average precision) for evaluating the accuracy of a bounding box in an object detection and segmentation model, FPS for detection of speed, and the metrics of the regression model of MAE (mean absolute error), MSE (mean square error), RMSE (root mean square error), and R^2 coefficient of determination, etc. Diversified evaluation indicators can help researchers evaluate and improve algorithms used in many aspects.

ROC curve is often used for evaluating two classifiers. The vertical axis of the ROC diagram is TPrate (Eq. 5) and the horizontal axis is FPrate (Eq. 6). FPrate represents the probability of misclassifying negative cases into positive cases, and TPrate represents the probability that positive cases can be divided into pairs. Each discrete classifier produces an (FPrate, TPrate) pair corresponding to a single point in ROC space. Several points in the ROC space are important to note. The lower left point (0, 0) represents the strategy of never issuing a positive classification; such a classifier commits no false positive errors but also gains no true positives. The opposite strategy of unconditionally issuing positive classifications is represented by the upper right point (1, 1). The point (0, 1) represents perfect classification (Fawcett, 2006).

$$TPrate = \frac{TP}{TP + FN} \tag{5}$$

$$FPrate = \frac{FP}{FP + TN} \tag{6}$$

In addition to ROC curve, MCC (Eq. 7) is also used to measure the performance of binary classification. This indicator considers true positives, true negatives, false positives, and false negatives. It is generally considered to be a relatively balanced indicator. It can be applied even when sample sizes of two categories are very different (Supper et al., 2007). MCC is essentially a correlation coefficient between actual classification and prediction classification, and its value range is [−1, 1]. When it is 1, it means perfect prediction of a subject; when it is 0, it means that the predicted result is worse than the random prediction result; −1 means that the predicted classification is completely inconsistent with the actual classification.

$$MCC = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \tag{7}$$

CONVOLUTIONAL NEURAL NETWORK-BASED FRESH FRUIT DETECTION

Fruit Flower Detection

Fruit flowers are the primary form of fruit organ. Most fruit trees bloom far more than final fruits. However, if there are too many flowers, nutrition supply will be insufficient, which will not only affect the normal development of fruits but will also cause formation of many small fruits and secondary fruits. Yield and economic benefits will be affected. Therefore, flower thinning is necessary to remove some excessive flowers and obtain high-quality fruits (Wouters et al., 2012). After flower thinning, flower detection is implemented and plays a considerable role in fresh fruit production. Flowers of most kinds of fruits are small and dense, resulting in overlap and blockage, which seriously affect the accuracy of detection. Precise estimation based on DL can assist orchardists in assigning labor resources on time to attain a highly effective but low-cost harvest.

The size of flowers of most species of fruits is small, and the flowers are dense, which causes overlap and occlusion quickly. Many researchers detect the flowers in outdoor fields close to make the most of flowers' traits. Being inspired by the performance of CNNs in computer vision tasks, Dias et al. (2018) incorporated CNN and SVM for apple flower detection. Lin et al. (2020) compared the performance of R-CNN, Fast-R-CNN, and Faster-R-CNN in recognizing strawberry flowers, and Faster-R-CNN ha higher accuracy (86.1%) than R-CNN (63.4%) and Fast-R-CNN (76.7%). Farjon et al. (2020) constructed a system for apple flower detection, density calculation, and flourish peak prediction. The detector in the system was based on Faster-R-CNN. Mask R-CNN with ResNeXt50 is a superior algorithm for recognizing citrus flowers and detecting their quality in an end-to-end model. The average precision of detecting citrus flowers is 36.3, and the error of calculating the number was decreased to 11.9% (Deng et al., 2020). Using U-Net (Ronneberger et al., 2015) as the backbone of Mask-Scoring-R-CNN can also detect flowers with great precision (Tian et al., 2020). At the same time, researchers augmented a data set based on apple flowers' growth and distribution features to improve the learning capacity of networks. YOlOv4 can detect objects on three different scales. Wu D. et al. (2020) proposed a channel-pruning algorithm based on the YOLOv4 model. The pruned model contains simple structures and has fewer parameters, and it works with sound accuracy and faster speed.

Grape flower counting is often very time-consuming and laborious because the grape flower has particular phenotypic traits that their shapes are the small sphere and growing on the inflorescence densely. Hence, scholars utilized full convolution net (FCN) to detect and identify inflorescences, and then used

CHT to recognize the flowers (Rudolph et al., 2019). Palacios et al. (2020) also detected inflorescences and flowers, but both steps used the SegNet architecture with a VGG19 network. In addition, they estimated the actual number of flowers from the number of detected flowers by training a linear regression model. Litchi flowers are also densely clustered and difficult to distinguish in morphology. Thus, a semantic segmentation net that constituted of a backbone net, DeepV3, for feature extraction and a full convolutional net for pixel prediction can detect litchi flower at the pixel level (Xiong et al., 2021).

Growing Fruit Detection

Terrestrial Platform

In addition to fruit flower detection, fruit detection and counting are also important for yield estimation. Fruit growth in fruit trees is different, and fruit thinning needs to be implemented to remove small fruits, residual fruits, diseased fruits, and fruits with incorrect shapes, so that fruits are evenly distributed in trees and branches and can fully receive nutrients. After the fruit thinning and fruit dropping stages, fruits can be detected during fruit ripening to estimate yield (Zhou et al., 2012).

The CNN algorithm has better performance for detecting expanding fruits in a vast scene, which has been proved by comparing it with existing methods (Bargoti and Underwood, 2017). Various species of fruits have different characteristics; therefore, different CNN models are used. Tu et al. (2020) proposed a MS-FRCNN model to estimate passion fruit production. To detect fruits of small and dense olive, researchers tested five different CNN configurations in an intensive olive orchard, and the model with Inception-ResNetV2 showed the best behavior (Aquino et al., 2020). Behera et al. (2021) proposed a Faster-R-CNN model with MIoU, and it achieved an F1 score of 0.9523 and 0.9432 for yield estimation of apple and mango in the ACFR data set. Janowski et al. (2021) employed the YOLOv3 network to predict the yield of an apple orchard. Nevertheless, all algorithms face the problems of occlusion resulting from leaves or branches and fruit overlap. To suppress the disturbance from occlusion, an instance segmentation neural net based on Mask-R-CNN was used to detect apples in two-dimensional space and a multi-view structure from motion (SFM) (Triggs et al., 2002) was used to generate a 3D point cloud according to 2D detection results. Recognizing unripe tomatoes is important for long-term yield prediction, but green fruits are hard to perceive in a green background. Mu et al. (2020) used Faster-R-CNN to detect immature tomatoes in greenhouses and created a tomato location map from detected images. Prediction errors of a whole orchard caused by duplicate statistics attracted the attention of many scholars. It is remarkably effective segmenting individual mango trees with LiDAR Mask and identifying fruits with a Faster-R-CNN-based detector. Koirala et al. (2019a) designed a mango identification system and installed it on a multifunctional agricultural car to realize real-time detection. The algorithm named "MnagoYOLO" in the detction system is modified based on YOLOv2. The car drove on the path between rows of mango trees while the system detected and summed the mangoes on the trees (Koirala et al., 2019a). Some researchers thought of using mobile phones to detect kiwifruits in an orchard in real-time (Zhou et al., 2020). They used a single shot multi-box detector (SSD) with two lightweight backbones, MobileNetV2 and InceptionV3, to develop a device for kiwifruit detection in the wild, the Android app KiwiDetector. Four types of smart phones are used for experiments. Highest detection accuracy can reach 90.8%, and fastest detection speed can reach 103 ms.

Deep learning has advantages in yield estimation of clustered fruits. For dense small fruits such as blueberries and small tomatoes, DL has a better detection effect on single fruits and is more convenient for counting fruits. However, using DL to detect small fruits is more vulnerable to the influence of light conditions. To quantify the number of berries per image, a network based on Mask R-CNN for object detection and instance segmentation was proposed by Gonzalez et al. (2019). Grapes are a type of crop presenting a large variability in phenotype. Zabawa et al. (2020) chose to train a CNN to implement semantic segmentation for single grape berry detection, and then used the connected component algorithm to count each berry. SfM (structure-from-motion) can simultaneously solve camera pose and scene geometry estimation to find a three-dimensional structure. Thus, Santos et al. (2020) used Mask-R-CNN to segment grape clusters and generate comprehensive instance masks. Then, the COLMAP SfM software can match and track these masks to reduce duplicate statistics. GPS was employed to establish pair-wise correspondences between captured images and trajectory data (Stein et al., 2016). **Figure 9** displays the process of instance matching and tracking. A counting method for cherry tomatoes based on YOLOv4 was proposed by Wei et al. (2021), and it takes the counting problem as detecting and classifying problems that can reduce the effects of occlusion and overlap. Ni et al. (2021) proposed a method for counting blueberries based on the result of individual 3D berry segmentations. In that study, Mask-R-CNN was used for 2D blueberry detection, and the 3D point was used for 3D reconstruction.

Some types of fruits are only edible when ripe. Therefore, maturity monition can provide a timely signal to harvest workers. Tomatoes have the characteristics of clustered growth and batch ripening. Immature tomatoes contain solanine, which is noxious to the human body. Thus, dozens of studies are related to tomato maturity detection. Sun et al. (2018) first used Faster-R-CNN with ResNet 50 to detect critical organs of tomatoes, and the mAP of the model is 0.907. Subsequently, they improved the FPN model to recognize tomato flowers, green tomatoes, and red tomatoes, and the mAP achieved 0.995 (Sun et al., 2020). Coconuts with different maturities can be sold for various purposes. Therefore, Parvathi and Tamil Selvi (2021) used Faster-R-CNN to detect the maturities of coconuts in trees to decrease economic loss. The definition of mature and immature fruits is the primary issue of maturity detection. Some researchers transformed the identification task into a classification task. According to the relationship between storage time and appearance, tomatoes can be classified into five categories: "Breaker," "Turning," "Pink," "Light red," and "Red." A CNN can classify the level of tomato maturity (Zhang L. et al., 2018). Tu et al. (2018) collected five maturities category pictures of passion fruit (**Figure 10**), and then modified the Faster-R-CNN model to recognize the fruit

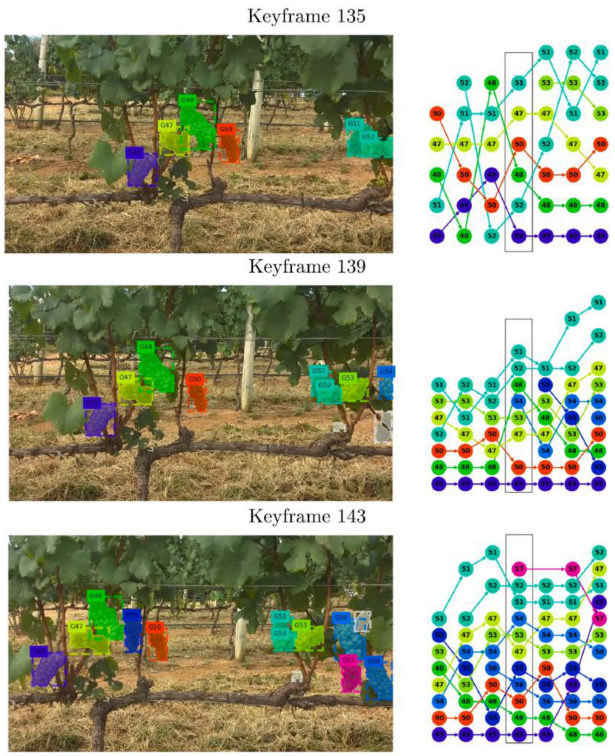

FIGURE 9 | Instance matching and tracking by 3-D assignment. **(Left)** Key frames extracted from a video sequence with a 1,080-p camera. **(Right)** Graph-based tracking. Each column represents instances found by a neural network, and each color represents an individual grape cluster in a video frame.

and its ripeness. Tian et al. (2019) divided objective apples into three classes, young, expanding, and ripe, and optimized the YOLOv3 model with DenseNet for detection. The classification method referred in Tian et al. (2019) was used on litchi (Wang H. et al., 2021). However, litchi fruits are different from apples that are small and dense; thus, Wang adjusted the prediction scale and decreased the weight layers of YOLOv3 to enhance the capacity of the model for compact object detection. Khosravi et al. (2021) coded olives according to their mature stages and varieties, divided them into eight categories, and used a deep convolutional network for detection. The overall accuracy of detection can reach 91.9, and the processing speed on the CPU is 12.64 ms per frame.

Offering indices of fruit maturity can help workers make harvesting plans and assist harvest robots in making decisions. Some scholars offered indices for describing fruit maturity under the premise of using a CNN to detect fruits. Huang et al. (2020) utilized Mask-R-CNN to identify the location of tomatoes in images and evaluated the HSV value of detected tomatoes. They then constructed Fuzzy inference rules between the maturity and the color feature of the surface of tomatoes, which can predict ripeness and harvesting schedule. Ni et al. (2020) also used Mask-R-CNN to extract blueberry fruit traits and gave two indices to describe fruit maturity (**Figure 11**). One index is about the maturity of individual berries that can infer whether blueberries are harvestable or not. Another is the maturity ratio (mature berry number/total berry number) of a whole cluster that can indicate the specific harvesting time of this cultivar. For clustered and dense fruits such as blueberries, cherries, and cherry tomatoes, the maturity of whole bunches of fruits can be calculated by detecting the maturity of each fruit using DL. At the same time, the labeling process is time-consuming and laborious. To provide technical support for high quality cherry production, Gai et al., 2021 proposed aYOLO-V4-dense model for detection of the maturity of cherries.

Aerial Platform

Many researchers have begun using UAVs (unmanned aerial vehicles) to obtain images, and UAVs have become common in agricultural remote sensing as intelligent devices progress. Studies have demonstrated that data taken with UAVs are suitable for fruit yield prediction (Wittstruck et al., 2021). Chen et al. (2017) proposed a novel method that uses DL to map from input images to total fruit counts. It utilizes a detector based on an FCN model to extract candidate regions in images, and a counting algorithm based on a second convolutional network that estimates the number of fruits in each region. Finally, a linear regression model maps that fruit count estimate to a final fruit count. A UAV-based visual detection technology for green mangoes in trees was proposed by Xiong et al. (2018). In their study, the YOLOv2 model was trained for green mango identification. The mAP of the trained model on the training set was 86.4%, and estimation error rate was 1.1%. Apolo-Apolo et al. (2020) used a UAV to monitor citrus in orchards (shown in **Figure 12**) and adopted Faster-R-CNN to develop a system that can automatically detect and estimate the size of citrus fruits and estimate the total yield of citrus orchards according to detection results. To solve the problem of inconvenient data capture in mountain orchards, Huang et al. (2022) designed a real-time citrus detection system for yield estimation based on a UAV and the YOLOv5 model. Kalantar et al. (2020) presented a system for detection and yield estimation of melons with a UAV. The system included three main stages: CNN-based melon recognition, geometric feature extraction (Kalantar et al., 2019), and individual melon weight (Dashuta and Klapp, 2018).

After using UAVs to predict fruit yield produced significant results, some scholars began to use UAVs to detect fruit maturity. Chen et al. (2019) used a UAV to capture images of the strawberry crop, and then utilized Faster-R-CNN to detect strawberry flowers and immature and mature strawberries with 84.1% accuracy. Zhou et al. (2021) also divided the growth of strawberries into three stages, "flowers," "immature fruits," and "mature fruits," and utilized the YOLOv3 model to detect images photographed with a UAV. The experimental results show that the model has the best detection effect on the data set taken with the UAV 2 m away from fruits, and the mAP reaches 0.88.

Differences Between Two Platforms

In Sections "Terrestrial Platform" and "Aerial Platform," we have described in detail the existing literature on the use of DL for

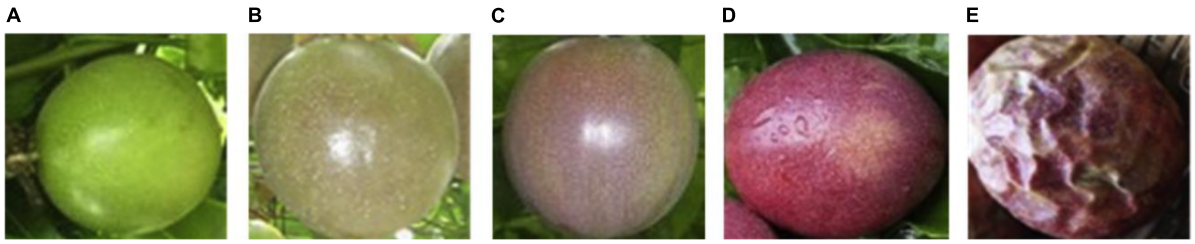

FIGURE 10 | Different maturity levels of passion fruit in Tu et al. (2018). **(A)** Near-young passion fruit. **(B)** Young passion fruit. **(C)** Near-mature passion fruit. **(D)** Mature passion fruit. **(E)** After-mature passion fruit.

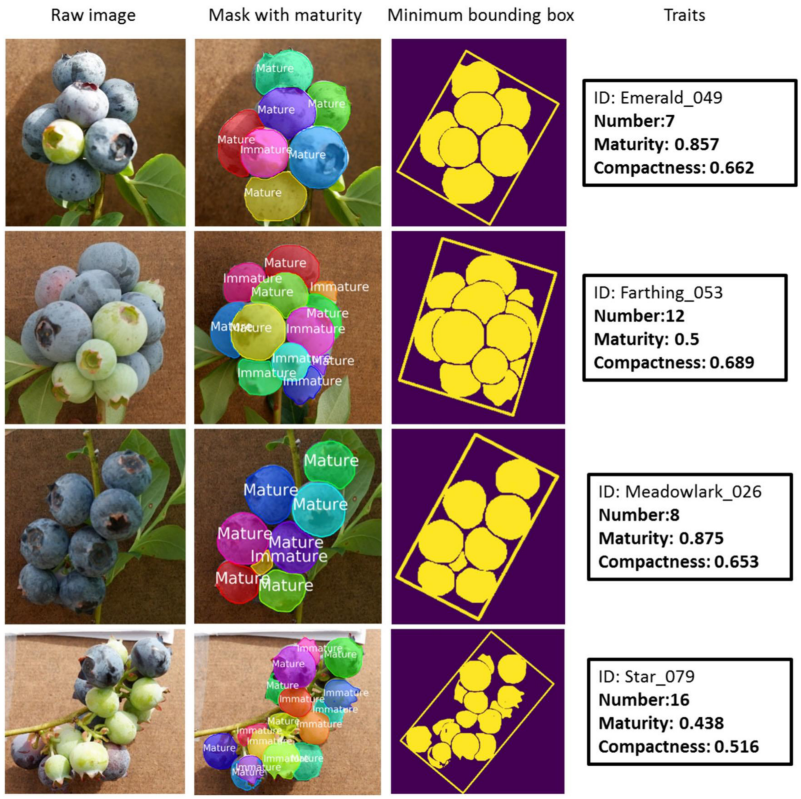

FIGURE 11 | Detection examples in Ni et al. (2020). The black rectangle contains the ID number and three traits (number, maturity, and compactness) of the corresponding sample.

detecting fruits in the growing period, and the differences can be seen in **Table 5**.

From the above discussion, the advantages and disadvantages of terrestrial and aerial platforms for yield estimation and maturity detection are obvious. For orchards located in harsh terrains, it is time-consuming and laborious that researchers use hand-held cameras to obtain data sets, and it is difficult to achieve automatic detection. Researchers only need to remotely control a UAV to easily acquire a large data set with different terrains and shooting distances, which is more convenient than handheld cameras. However, a UAV cannot be too close to the detected subject in the air; otherwise, a collision accident will occur. Therefore, it is noticed that the operation of a UAV needs more skilled technology.

For the yield prediction task, a UAV can capture a wider field of vision, such as fruits at the top of trees. However, when a UAV is used for long-distance shooting, the visibility of fruits is low because fruits at the bottom or inside of a canopy cannot be recognized, and increase in prediction error. When a handheld camera is used, the visibility of fruits is higher because a small part of a blocked fruit can be detected. However, the repetition rate of photographed fruits is high, which is not conducive to yield estimation.

For the maturity detection task, the characteristics of fruits are more conspicuous when a handheld camera is used for

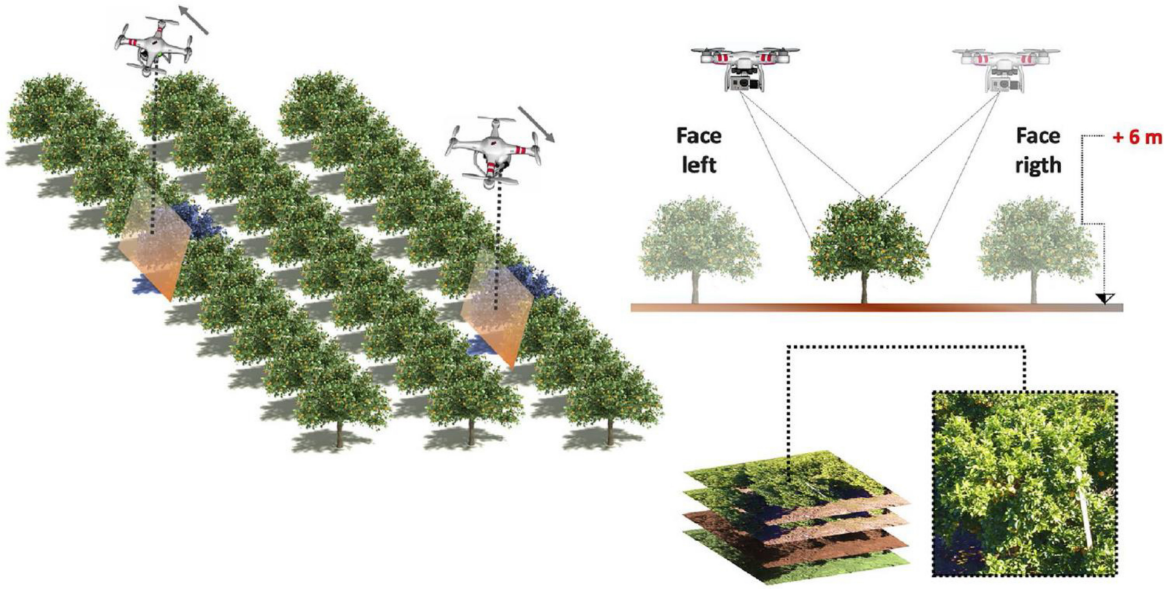

FIGURE 12 | Workflow of field tests (Apolo-Apolo et al., 2020).

close shooting. Fruits photographed with the UAV equipment are too small because of long distance, and the characteristics are relatively fuzzy. In Zhou et al. (2021), researchers used UAV equipment and a handheld camera equipment for data acquisition. They divided the strawberry data captured with the camera into seven different growth stages: flower fruits, green fruits, green-white fruits, white-red fruits, red fruits, and rotten fruits. The strawberry data collected with the UAV were only divided into three labels: flowers, immature fruits, and mature fruits.

Fruit Picking

The picking period of fruits arrives when fruit organs expand to a certain size. Mature fruits are needed to harvest fruits in time. However, there has been an imbalance between labor force and economic benefits for a long time. In these years, automatic fruit harvest robots have become a hotspot of intelligent agricultural study. Most fruit trees have proper growth heights and structured planting modes that offer convenience to harvest robots. **Table 6** summarizes the crops (containing fruits, branches, and trunks) experimented on for automatic harvest and corresponding detection models.

Fruit Recognition on Fields

The recognition and detection of fruits in an orchard environment provide robots with vital contextual information for maneuvering. However, branches, foliage, and illumination conditions affect the fruit detection with robots. Feature augmentation is a simple way to enhance the learning capacity of DL models. Mu et al. (2019) collected images with four types of occlusions in four illumination conditions as training data. Some researchers divided target apples into four classes depending on their obscured circumstances: leaf-occluded, branch/wire-occluded, non-occluded, and occluded fruits (Gao et al., 2020). Different varieties of the same fruit will have subtle differences in appearance. Using Mask-R-CNN to segment fruit images can distinguish fruits from occluded ones well. Chu et al. (2021) used an integrated data set with two varieties of apple to train Mask-R-CNN for suppression. Jia et al. (2020) optimized the Mask-R-CNN model in the backbone net, ROI layer, and FCN layer for apple harvesting robots. In a research study on strawberry harvest, the researchers reduced the magnitude of backbone and mask network and used a process of filtering and grouping of candidate regions to replace the object classifier and the bounding box regressor Mask-R-CNN. The new architecture can process original high-resolution images at 10 frames per second (Pérez-Borrero et al., 2020). Then, Pérez-Borrero et al. (2021) proposed a new strawberry instance segmentation model based on FCN whose FPS rate was six times higher than those obtained in reference methodologies based on Mask R-CNN.

As we have discussed in Section "Dataset Acquisition," a depth image contains more information. Ganesh et al. (2019) assessed the performance of Mask-R-CNN by applying three forms of color space input, RGB images, HSV images, and RGB + HSV images. The result showed that adding HSV information to RGB images can decrease false positive rate. Sa et al. (2016) explored two methods for imagery modality fusion based on Faster-R-CNN. One is early fusion (**Figure 13A**) by augmenting channels of input images from three (red, green and blue) to four (red, green, blue, and NIR) channels. Another is later fusion (**Figure 13B**) that fuses pieces of classified information of an RGB-trained model and an NIR-trained model. NIR (near infrared) here refers to images

TABLE 5 | Summary of related studies on application of CNN-based detection models in growing fruits.

Platform	Purpose	Detected object and label	CNN-based detection model	Following-up works	Remarks	References
Terrestrial platform	Yield estimation	Apple	Mask-R-CNN (2D detection)	SFM photogrammetry is used for generating 3D point cloud and SVM is used for removing false positive	Detection accuracy: 76.2% (2D image detections) and 85.7% (3D detections). Prediction precision: R^2 = 0.8	Gené-Mola et al., 2020
		Apple	YOLOv3	Counting detected fruits for yield estimation	Detection accuracy: 84%	Janowski et al., 2021
		Mango	Faster R-CNN	The GPS/INS, color cameras with strobes, and LiDAR used for fruit locating, tracking, and counting	Prediction accuracy: R^2 = 0.94	Stein et al., 2016
			MangoYOLO	Correction factors are used for estimating yield load	Detection precision: 98.3%. Estimation precision: 4.6–15.2% of packhouse fruit counts	Koirala et al., 2019a
		Tomato	Faster R-CNN	Stitching detected images and compiling a tomato location map of a greenhouse, estimating tomato size as per bounding box size.	Model performance: average precision: 87%, R^2 = 0.87	Mu et al., 2020
		Cheery tomato clusters	YOLOv3	ResNet-50 is used for classifying fruit clusters and counting total fruit number	Prediction precision: RMSE = 6.37, MAPE = 13.9%	Wei et al., 2021
		Passion fruit	Faster R-CNN	Counting detected fruits for yield estimation	Model performance: P = 96.2%, R = 93.1%, F1 = 0.95	Tu et al., 2020
		Oliver	Inception-ResNetV2	Counting detected fruits for yield estimation	Model performance: F1 = 0.84	Aquino et al., 2020
		Grape clusters	MobileNet-V2	DeepLabV3 segmenting each berry for counting	Berry detection accuracy of 94.0% in the VSP and 85.6% in the SMPH	Zabawa et al., 2020
		Kiwifruit	SSD (with MobileNetV2, quantized MobileNetV2, InceptionV3, and quantized InceptionV3)	Performing on mobiles with Android system and counting detected fruits for yield estimation	True detected rate (TDR) of MobileNetV2, quantized MobileNetV2, InceptionV3, and quantized InceptionV3 are 90.8%, 89.7%, 87.6%, and 72.8%, respectively.	Zhou et al., 2020
		Blueberry	Mask-R-CNN	Using different backbones: ResNet101, ResNet50 and MobileNetV1 to Mask-R-CNN and adding a step to outputs each instance of a blueberry to quantify the total number of blueberries in an image.	The best result was obtained when the ResNet50 backbone was used achieving a mIoU score of 0.595.	Gonzalez et al., 2019
		Blueberry	Mask-R-CNN	3D minimum bounding box calculating fruit cluster compactness after 3D reconstruction and proposing a trait extraction algorithm to segment individual 3D blueberries, count berry number, calculate maturity, and estimate berry size.	The average counting accuracy for the 40 samples is 97.3%. The fruit clusters with a low fruit number generally have a higher accuracy, resulting in almost 100% accuracy.	Ni et al., 2021
		Multi-fruit	Faster-R-CNN with MIoU	Counting detected fruits for yield estimation	Model performance: R^2 of mango, pomegranate, tomato, apple & orange are 0.98, 0.92, 0.96, 0.98, and 0.95	Behera et al., 2021
	Maturity detection	Apple ("Young Apple," "Expanding apple," "Ripe apple")	YOLOv3	Using different data augment methods and data numbers to comparison. Detection under occlusion and overlapping apple conditions and no apple environment.	Model performance: F1 = 0.817. Average detection time: 0.304 s	Tian et al., 2019
		Tomato ("Flower," "Green tomato," "Red tomato")	Faster-R-CNN	Taking comparison between YOLOv2, YOLOv3, original Faster-R-CNN, R-FCN, and proposed model.	Model performance: Mean average precision: 90.7%. Average test time: 0.073 s. Model memory: 115.9 MB	Sun et al., 2018
		Tomato ("Breakers," "Turning," "Pink," "Light red," "Red")	Own model	Using own designed CNN for images classification	Classification accuracy: 91.9%	Zhang L. et al., 2018

(Continued)

TABLE 5 | (Continued)

Platform	Purpose	Detected object and label	CNN-based detection model	Following-up works	Remarks	References
		Tomato ("Immature," "Breaker," "Preharvest," "Harvest")	Fuzzing Mask-R-CNN	Locating the stalk points of ripe tomatoes by obtaining the contours of tomatoes from Mask-R-CNN for harvesting.	Model performance: P = 96.1%, R = 95.9%.	Huang et al., 2020
		Four blueberry cultivars ("Immature" and "Mature")	Mask-R-CNN	Defining and calculating blueberry maturity and compactness. Assessing the extracted traits and delineating trait differences in four blueberry cultivars.	Model performance: Mean average precision: 78%. R^2 of four cultivars: 0.932, 0.877, 0.859, 0.934.	Ni et al., 2020
		Coconut ("coconut" and "Mature coconut")	Faster-R-CNN	Comparing the performance of Faster-R-CNN with different backbones, comparing the performance of improved model and other objection detection models.	Model performance: Mean average precision: 89.4%. Detection speed: 3.124 s	Parvathi and Tamil Selvi, 2021
		Passion fruit ("After-mature," "Mature," "Near-mature," "Near-young," "Young")	Faster R-CNN	Using DSIFT algorithm and LLC algorithm to extract the features of fruit from R, G, B channels and send the representative features to SVM classifier for maturity indentation.	Detection accuracy: 92.71% and maturity classification accuracy: 91.52%	Tu et al., 2018
		Litchi ("Ripe litchi," "Expanding litchi," "Young litchi")	YOLOv3-Litchi	Comparing the proposed model with YOLOv2, YOLOv3, and Faster-R-CNN.	Model performance: average detection time: 0.029 s, mean average precision: 75.6%, average precision of young litchi, expanding litchi, and expanding litchi is 67.3%, 71.9%, 73.8%.	Wang H. et al., 2021
		Oliver ("ZIG," "RIG," "ZVS," "RVS," "ZBS," "RBS," "ZOR," "ROR")	Own model	Evaluating the efficiency of six optimizers: Adagrad, SGD, SGDM, RMSProp, Adam, and Nadam.	Overall accuracy 91.91%, detection speed: 12.64 ms/frame (CPU)	Khosravi et al., 2021
		Strawberries ("Flower," "Flower-Fruit," "Green-Fruit," "Green-White-Fruit," "White-Red-Fruit," "Red-Fruit," and "Rotted-Fruit")	YOLOv3	Identify the different ripeness of the detected fruit.	The mAP of strawberry maturity classification was 0.89, and the highest classification AP was 0.94 for fully matured fruit.	Yue et al., 2020
		Cherry ("Cherry," "Cherry_1," "Cherry_2")	YOLOv4	DenseNet is used to replace the CSPDarkNet53 in YOLO-V4 and comparing different models in detecting ripe cherries	The mAP increased 0.15 comparing with the YOLO-V4 model and the F1 scores, IOU is 0.947 and 0.856.	Gai et al., 2021
Aerial platform	Yield estimation	Apple, orange	FCN	A second neural network and a linear regression were used to count the number of fruit.	Mean IU of 0.813 on the oranges and 0.838 on the apples, a best l2 error of 13.8 on the oranges, and 10.5 on the apples	Chen et al., 2017
		Green mango	YOLOv2	Counting detected fruits for yield estimation.	The mAP was 86.4%, a precision was 96.1% and a recall rate was 89.0%.	Xiong et al., 2018
		Citrus	Faster-R-CNN	Counting detected fruits and estimate the weight for yield estimation.	Mean error is 7.22%.	Apolo-Apolo et al., 2020
		Citrus	YOLOv5	Comparing the proposed model with different models and different occlusion degrees.	Accuracy: 93.32%, speed: 180 ms/frame, FPS: 83 s (In 2080ti), recall: 88.78%	Huang et al., 2022
		Melon	RetinaNet	Estimate the weight of the detected fruit.	Overall average precision score: 0.92 and F1 is more than 0.9	Kalantar et al., 2020
	Maturity detection	Strawberries ("Flower," "Immature Fruit," "Mature Fruit")	YOLOv3	Identify the different ripeness of the detected fruit.	For Flower, Immature Fruit, and Mature Fruit detection from the test data set at 2 m, the APs were 0.83, 0.87, and 0.93, the mAP for the test data set at 2 m was 0.88.	Zhou et al., 2021

taken by near-infrared imaging technology. There are also two fusion methods for detecting kiwifruits based on Faster-R-CNN (Liu et al., 2019). One is similar to the early fusion (Sa et al., 2016), and the other fuses the feature maps from two modes displayed in **Figure 14**. The background objects of RGB-D images captured with a Kinect V2 camera can be

TABLE 6 | Summary of related studies on application of CNN-based detection models in fruit harvesting.

Crop applied	Basic model	Data augment	Dataset	Transfer learning	Detection rate (%)	Inference speed (s/image)	References
Apple	SSD	√	589 RGB images	√	89.2	–	Peng et al., 2018
	R-CNN	–	270 RGB-D images	√	86.0	–	Zhang J. et al., 2018
	Faster-R-CNN	√	967 three-modalities images (RGB, range-corrected intensity, and depth)	√	94.8	0.074 @548×373 px	Gené-Mola et al., 2019a
	SSD	–	250 RGB-D images	–	92.3	2.00 @3840×1080 px	Onishi et al., 2019
	LedNet (FPN+ASPP)	√	1,100 RGB images	√	85.3	0.028 @320×320 px	Kang and Chen, 2020
	Faster-R-CNN	√	12,800 RGB images	–	87.6	0.241 @1920×1080 px	Gao et al., 2020
	Faster-R-CNN	√	800 RGB-D images	√	87.1	0.124 @1920×1080 px	Fu et al., 2020b
	Faster R-CNN	√	820 RGB images	–	92.5	0.058 @100×100 px	Wan and Goudos, 2020
	Faster-R-CNN	√	675 RGB-D images	√	82.4	0.450 @360×640 px	Zhang et al., 2020
	Mask-R-CNN	–	1,140 RGB images	√	97.3	–	Jia et al., 2020
	Mask-R-CNN	√	24,005 RGB images	√	58.1	–	Dong W. et al., 2021
	Mask-R-CNN	–	19,528 RGB images	–	88.0	0.250 @1280×720 px	Chu et al., 2021
	DenseNet+FPN	√	953 RGB images	√	93.2	0.023 @200×308 px	Xu et al., 2021
Citrus	SSD	√	1,660 RGB images	√	91.1	–	Peng et al., 2018
	Mask-R-CNN	–	300 RGB images	√	85.1	0.045 @1024×768 px	Liu Y. P. et al., 2018
	Mask-R-CNN	√	RGB and RGB-HSV images	√	97.5	0.011 @256×256 px	Ganesh et al., 2019
	Mask-R-CNN	–	5,195 RGB images	√	–	–	Xiong et al., 2019
	Mask-R-CNN	–	750 RGB images	–	98.2	0.700 @1024×768 px	Yang et al., 2019
	Mask R-CNN	–	5,195 RGB images	–	92.2	9.230 @1920×1080 px	Yang et al., 2020
	Faster R-CNN	√	799 RGB images	–	90.7	0.058 @100×100 px	Wan and Goudos, 2020
Kiwifruit	Faster-R-CNN	–	20,160 images	√	92.3	0.274 @2352×1568 px	Fu et al., 2018
	Faster-R-CNN	√	20,160 images	√	87.6	0.347 @2352×1568 px	Song et al., 2019
	Faster-R-CNN	√	21,147 RGB images	√	96.0	1.070 @1920×1080 px	Mu et al., 2019
	Faster-R-CNN	–	1,000 NIR images+1,000 RGB images+1,000 RGB-D images	–	91.7	0.134 @512×424 px	Liu et al., 2019
	YOLOv3	√	20,160 RGB images	√	90.1	0.034 @2352×1568 px	Fu et al., 2021
Strawberry	SSD	√	4,550 RGB images	√	87.7	0.23 @360×640 px	Lamb and Chuah, 2018
	Mask R-CNN	–	2,000 RGB images	√	95.8	0.125 @640×480 px	Yu et al., 2019
	Mask R-CNN	√	–	–	81.0	0.620 @640×480 px	Ge et al., 2019
	Mask R-CNN	–	–	–	–	–	Xiong et al., 2020
	Mask R-CNN	√	3000 RGB images	–	78.3	0.01 @ 1008×756 px	Pérez-Borrero et al., 2020
	FCN		3100 RGB images	–	93.4	0.03 @ 1008×756 px	Pérez-Borrero et al., 2021
Grape	Mask R-CNN	√	1,050 RGB-D images	–	89.5	1.100 @1920× 1080 px	Yin et al., 2021
Litchi	SSD	√	636 RGB images	√	86.7	–	Peng et al., 2018
Mango	Faster R-CNN		822 RGB images	–	88.9	0.058 @100×100 px	Wan and Goudos, 2020
	DenseNet+FPN	√	1694 RGB images	√	93.6	0.023 @500×500 px	Xu et al., 2021
Rosa roxburghii	Faster R-CNN	√	8,475 RGB images	–	92.0	0.200 @500×500 px	Yan et al., 2019
Guava	Mask R-CNN	√	304 RGB-D images	√	53.7	0.250 @512×424 px	Lin et al., 2021
Sweet pepper	Faster-R-CNN	√	122 RGB-NIR images	√	83.8	0.393 @–	Sa et al., 2016
	Deep CNN	√	960 RGB images	–	82.9	–	Rehman and Miura, 2021
Cherry tomato	YOLOv3	√	1825 RGB images	–	96.8	0.058 @ 1,292×964 px	Chen et al., 2021

filtered by distance threshold and foreground-RGB images, and Faster-R-CNN with VGG achieved a high average precision of 0.893 for the foreground-RGB-images (Fu et al., 2020b). Gené-Mola et al. (2019b) added an imaging modality, the range-corrected IR intensity proportional to reflectance, based on RGB-D images. It makes an input image become five channels, and the F1-score of the detection model improves 4.46% more than simple RGB images.

In most studies, researchers spent energy optimizing algorithms. Peng et al. (2018) used SDD and replaced the original VGG-16 with ResNet-101 to detect apple, citrus, and lichi. Besides, decreasing layers of the backbone of SSD can achieve

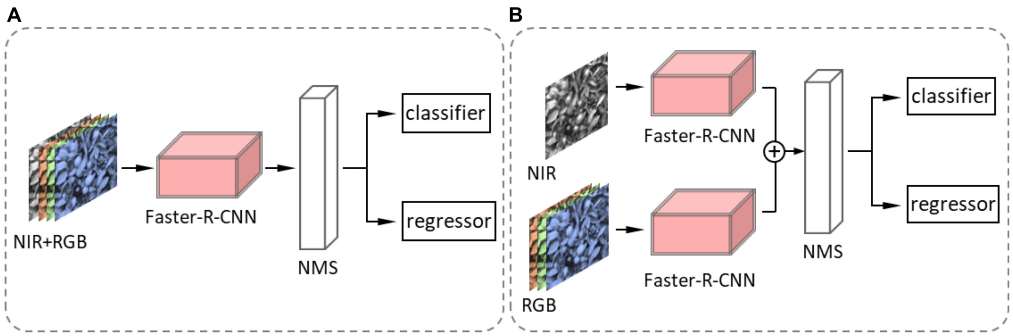

FIGURE 13 | Diagram of fusion methods in Sa et al. (2016). **(A)** Early fusion: first, channels of the detected image are augmented from three to four channels. Second, the augmented image is detected by Faster-R-CNN. Third, NMS (non-maximum suppression) removes duplicate predictions. Finally, the classifier and regressor calculate the category and coordinate of the bounding box. **(B)** Late fusion: first, the RGB image and the NIR image are detected by Faster-R-CNN. Second, the detected outputs from two Faster R-CNN networks are fused. Third, NMS (non-maximum suppression) removes duplicate predictions. Finally, the classifier and regressor calculate the category and coordinate of the bounding box.

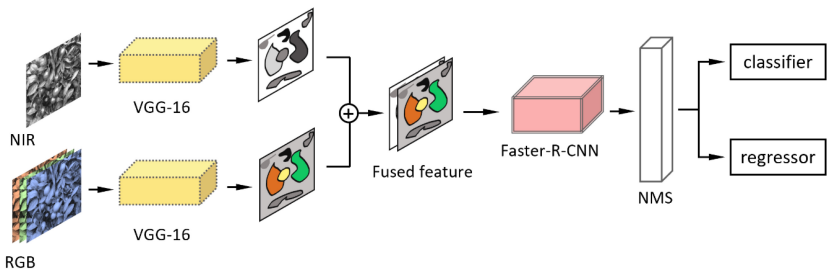

FIGURE 14 | Feature-fusion model in Liu et al. (2019). First, it inputs the RGB and NIR images separately into two VGG16 networks and then combined them on the feature map; then, the feature map is detected by Faster-R-CNN.

accurate and precise detection in a low-power hardware (Lamb and Chuah, 2018). Kang and Chen (2020) designed a CNN model named "LedNet," which is mainly improved by a lightweight backbone, FPN, and ASSP, for fruit detection in an apple orchard. Integration of DenseNet and FPN can obtain small fruits' features more correctly (Xu et al., 2021). Fu et al. (2018) first used a DL model for kiwifruit detection in 2018, and they developed a kiwifruit detection system based on Faster-R-CNN with ZFNet for filed images. Three years later, they proposed a DY3TNet model based on the addition of convolutional layers to YOLOv3-Tiny for kiwifruit recognition in a wild environment (Fu et al., 2021). Some scholars are also dedicated to kiwifruit detection but used Faster-R-CNN with VGG-16; however, the precision and speed of detection are lower than the results of Fu et al. (2018). Modification of the pooling layer can also improve detection accuracy. Yan et al. (2019) changed the Faster-R-CNN model by replacing the ROI pooling layer with the ROI align layer. Wan and Goudos (2020) modified the pooling layers and convolution layers of the existing Faster-R-CNN. In the two experiments (Yan et al., 2019; Wan and Goudos, 2020), detection speed and accuracy accomplished prominent improvements. As we know, most fruits are elliptical in a 2D space. Thus, specialists presented an ellipse regression model based on Mask-R-CNN for detecting elliptical objects and inferring occluded elliptical objects (Dong W. et al., 2021). The original YOLOv3 has low precision in detecting cherry tomatoes, and DPNs (dual-path networks) can extract richer features of recognition targets. Therefore, researchers improved the YOLOv3 model based on DPNs for identification of cherry tomatoes.

Obstacle Avoidance

Robots should also learn to avoid foliage and branches except when identifying fruits. For sure, researchers thought of making robots recognize obstructions while detecting fruits, so robots can react differently according to different objects. Using the R-CNN model to detect and locate branches of apple trees in natural environments can establish a branch of skeletons, so that the arms of robots can avoid branches while grabbing apples (Zhang J. et al., 2018). For citrus harvest, Yang et al. (2019) utilized the Mask-R-CNN model to recognize and reconstruct branches of citrus trees. Later, the researchers designed a recognition model based on their previous studies for citrus harvest robots to detect fruits and branches simultaneously (Yang et al., 2020). Lin et al. (2021) used a tiny Mask-R-CNN model to identify fruits and branches of guava trees and reconstructed the fruits and branches for robotic harvest.

There are some other means for occlusion avoidance except when detecting obstructions. Rehman and Miura (2021) presented a viewpoint plan for fruit harvest. They demonstrated the possible types of a fruit in one scene with the labels "center,"

"left," "right," "occluded," which are depicted in **Figure 15**. The arm of a robot is qualified to determine the harvesting path as per detected labels. What is more, objective fruits could be classified into normal, branch occlusion leaf occlusion, slight occlusion overlapping, or main branch (Liu Y. P. et al., 2018). Also, a new strawberry-harvesting robot with a more sophisticated active obstacle separation strategy has been developed, and the strawberry location detector in the system is based on Mask-R-CNN (Xiong et al., 2020).

Picking Point Detection

The feasibility of automatic harvesting has been confirmed broadly. A further important issue is locating harvesting points precisely that can guarantee that the robot's grasp of fruits is accurate and uninjurious. Mask-R-CNN not only can detect an object accurately but can also generate corresponding masks of an object region at the pixel level, which can assist in locating picking points. Longye et al. (2019) segmented and reconstructed the overlapping citrus using the Mask-R-CNN model and performing concave region simplification and distance analysis. Strawberry detection can also employ the Mask-R-CNN model. Then, picking points are determined by analyzing the shape and edge of objective masks (Yu et al., 2018). Ge et al. (2019) also utilized the Mask-R-CNN model to detect strawberries based on RGB-D images that have depth information of images; they performed coordinate transformation and density-based point clustering, and proposed a location approximation method to help robots locate strawberry fruits. Yin et al. (2021) proposed segmenting the contours of grapes from RGB images with Mask-R-CNN and then reconstructing a grape model by fitting a cylinder model based on point cloud data extracted from segmented images. By recognizing and calculating the outline of a bunch of grapes, the arm of robot can grab stalks at the top of a bunch of grapes. Shake-and-catch harvesting first appeared in 2010 (He L. et al., 2017). Some researchers used the Faster-R-CNN model to establish a relationship between fruit location and branch location (Zhang et al., 2020). Connections can help a robot to determine shake points.

Generally, researchers detect fruits on the side of trees, but Onishi et al. (2019) proposed a novel method for inspecting apples from below. The SSD model is used to detect the 2-D position of the apple shown in **Figure 16A**. The stereo camera ZED provides the 3-D position of the center of the bounding box, which is like in **Figure 16B**, and the position can be a picking point. Then, the robot can move below the target apple to grasp the fruit according to the predicted position like in **Figure 16C**.

Fruit Grading

After a fruit is picked, it will gradually flow to the market and produce economic benefits. Recently, customers have higher requirements for fruit quality as consumption levels increase. Hence, it is necessary to evaluate the quality of fruits before delivering them to consumers because of external and internal vulnerabilities. Those with better fructifications can be consumed, and those with worse can be processed to make fruit foods. Graded-based vendition by detecting internal diseases, sugar content, surface damages, maturity, size, etc. can promise both seller and purchaser benefits. In this section, we will introduce the research on CNN-based fresh fruit grading from grading as per external traits, grading as per internal traits, and fruit cultivar classification.

External Trait-Based Grading

External phenotypic characteristics of fruits directly show their qualities, which affect the sale price and consumer enthusiasm. Thus, external quality detection plays a significant role in fruit grading. Many experiments testified that CNNs have noteworthy superiority in fruit quality grading (Wang et al., 2018; Jahanbakhshi et al., 2020; Patil et al., 2021). In the research of Wang et al. (2018), a modified AlexNet model was used to extract the feature of defects on litchi surface and classify litchi defect images. The classification precision of the AlexNet-based full convolutional network is higher than that of linear SVM and Naive Bayes Classifier. Jahanbakhshi et al. (2020) compared sour lemon detection performance based on a CNN model with other image categorization methods and demonstrated the superiority of the CNN-based model in fruit grading. Patil et al. (2021) also concluded that CNNs have a faster speed of operation in dragon fruit grading and sorting by comparing the performance of ANN, s, and CNN models.

Apple is the most salable and lucrative fruit globally. Some researchers developed apple defect detection systems for apple grading. Fan et al. (2020) designed a 4-lane fruit sorting system to detect and sort defective apples, and a CNN model for a defective apple sorting system, in which a global average pooling layer was applied to replace a fully connected layer. Wu, Zhu, and Ren performed laser-induced light backscattering imaging to capture apple defect images and designed a simple CNN model to classify

FIGURE 15 | Possible types of fruit in one scene formulated by Rehman and Miura (2021). **(A)** Center, **(B)** left, **(C)** right, and **(D)** occluded.

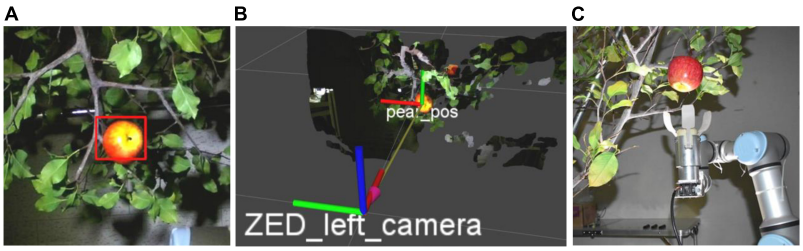

FIGURE 16 | Automatic apple harvesting mode in Onishi et al. (2019). **(A)** Detection of a two-dimensional position, **(B)** detection of a three-dimensional position, **(C)** approaching the target apple.

scabs on apple surface (Wu A. et al., 2020). Aside from scabs on apple surface, the CNN model can classify images of apples with bruises, cracks, and cuts (Nur Alam et al., 2020). Researchers also conducted related studies on other fruits. Azizah et al. (2017) used a CNN model to implement mangosteen surface defect detection. Zeng et al. (2019) constructed an ensemble-convolution neural net (E-CNN) model based on the "Bagging" learning method for detection of defects in jujube fruits. Cherries are prone to abnormal shapes during growth, so some researchers used a modified AlexNet model to classify cherries according to growth shapes (Momeny et al., 2020). Wu S. et al. (2020) combined and investigated several deep learning methods for detecting visible mango defects and found that VGG-16 has a dominant position by combining and investigating several DL methods. De Luna et al. (2019) also demonstrated that the VGG-16 model has better performance in tomato defect inspection. Some researchers used a modified ResNet-50 model to extract the features of tomato surface defects and classify images of tomato defects (Da Costa et al., 2020). Chen et al. (2021) established an online citrus sorting system, shown in **Figure 17**, and a detector named Mobile-citrus based on Mobile-V2 to identify surface defects in citrus. Then, the arms of robots arms pick out the defective ones with the linear Kalman filter model used in predicting the future path of the fruits.

The external appearance of a fruit sometimes also represents its freshness. A multi-class classifier based on VGG-16 and Inception-V3 was built by Ashraf et al. (2019) for detecting fresh and rotten fruits. Researchers also practiced the advantages of CNNs in classifying the freshness of apples, bananas, and oranges (Ananthanarayana et al., 2020).

Internal Trait-Based Grading

Commonly used RGB images cannot acquire internal traits of fruits, for instance, diseases, sugar content, moisture, etc. Consequently, many researchers combined CNN-based DL models with spectrum techniques and made remarkable progress in internal quality-based grading. The sweetness, crispiness, and moisture of apples can be detected using hyperspectral images and 3D-CNN (Wang et al., 2020). Researchers have also proposed a multi-task model based on 3D-CNN for predicting the sugar content and hardness of yellow peaches simultaneously (Xu et al., 2020). Jie et al. (2021) proposed a non-destructive determination method based on the YOLOv3 algorithm, and hyperspectral imaging technology contraposes citrus granulation.

CHALLENGES AND FUTURE PERSPECTIVE

As per the above statements, the appearance of CNN models is already invigorating the automatic production of fresh fruits. However, people remain having quite a lot of challenges to face, because the whole automation of the fruit industry is merely in the period of development.

Environmental Issues

The problem of fruits being occluded is a difficulty in fruit detection. Most occlusions are caused by foliage, branches, trunks, and fruit overlapping in complex fruit-growing environments. Moreover, varying illumination conditions are also one of the instability factors in fruit detection. For

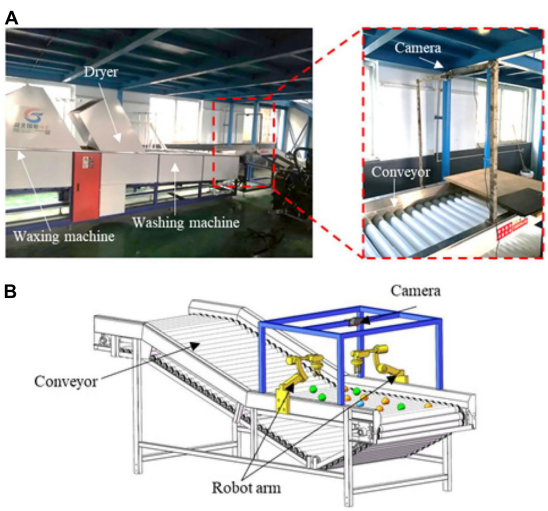

FIGURE 17 | Platform setup and computer vision system (Chen et al., 2021). **(A)** The citrus processing line was assembled in the laboratory, with a webcam mounted above the conveyor. **(B)** The diagram shows an automated citrus sorting system using a camera and robot arms, and the robot arms will be implemented in future studies.

instance, green fruits, such as green citrus, green litchi, avocado, and guava, conceal in a green background, which results in more faulty detections of machine visions. Thus, algorithms with high detection accuracy and speed are the objective of researchers.

In addition to algorithm improvement, human intervention can also assist in solving environmental issues. It is a feasible method to increase the visibility of fruits by trimming the crown of fruit trees and standardizing planting according to the principles of horticultural operations. For example, a trellised fruiting wall is suitable for robotic operations during pruning and harvesting (Majeed et al., 2020). Artificially improving the lighting of an environment can also reduce uncertainty in the process of detection. When light is strong, cameras are prone to overexposure. In response to this problem, some researchers have adopted a shading platform to reduce the impact of sun exposure (Gongal et al., 2016; Nguyen et al., 2016; Silwal et al., 2016). To increase the utilization rate of machines, people will have to let robots work at night. However, there is insufficient lighting during night operations, and external light sources are needed to improve the lighting of an environment (Koirala et al., 2019a). Most of the current shading devices and light supply devices are relatively bulky, so it is of commercial value to design a shading or a lighting system that is simpler and more portable.

Exploration of New Areas

In the process of fresh fruit production from blooming to marketing, and pollination, pesticide application, harvesting, sorting, and grading all need a large pool of workers. The preceding discussion suggests that most applications of CNNs in fresh fruit production are in the algorithm development stage. Autonomous operation of robots is mostly used for fruit harvesting and grading. There are fewer exploitations of automatic pollination robots for the problem of greenhouse plants' insufficient pollination. In current studies, Chunjiang Zhao utilized the improved YOLOv3 network to identify tomato flowers in greenhouses and embedded the system in automatic pollination robots. Phenology distribution monitoring can govern the timing and dosage of chemistry thinning, which determines the quality of fruits. Fruit flower phenology involves a period from the emergence of fruit buds to petal withering means that monitoring of flower phenology is not only estimating flower number. Studies on using computer vision to detect fruit flower phenology are rare, and CNN-based methods are even less. According to our search, Wang X. et al. (2021) designed a phenology detection model based on a CNN named DeepPhenology to estimate apple flower phenology distribution. Currently, more researchers are utilizing CNN to detect fruit flowers and achieve the purpose of yield estimation. Perhaps the application of CNN in fruit flowers phenology estimation is a new area worth exploring.

Food safety is an issue that concerns people, because accumulation of pesticides in the human body risks causing cancers. However, pesticide residues on fruit surfaces are inescapable, because orchardists will perform pesticide delivery to guarantee fruit's healthy growth. CNNs can be used to identify pesticide residues, but the CNN used in most studies (Yu et al., 2021; Zhu et al., 2021) is a one-dimensional CNN, and input data are pre-processing data extracted with a spectrometer. The process of detection is complicated and cumbersome. Rarely have researchers used the 2D CNN model to detect pesticide residues in harvested fruits (Jiang et al., 2019). Although pesticide residues belong to the external characteristics of fruits, its vision detection still needs hyperspectral images, because RGB images cannot capture pesticide residues. The current detection methods have complex processes out of proportion to the economic benefits generated by pesticide residue detection. Thus, the feasibility of using CNNs to detect pesticide residues in fruits should be studied further. When grading and sorting clustered fruits such as grapes, litchis, and longan, a manipulator grabs the stalk on the top of a fruit to minimize damage to the fruit. However, fruits on the sorting table are arranged disorderly, and stalks are not arranged neatly on a horizontal plane. Therefore, it is necessary to use CNNs to determine the robot's sequence of grabbing of clustered fruits (Zhang and Gao, 2020).

There is no doubt that CNNs have a developing potential in fresh fruit production. In future studies, it is promising to enhance the application areas of CNNs in fresh fruit detection. It could be a good direction that infuses CNNs into whole fruit production.

Execution of Multiple Tasks

Fruit surfaces are easily damaged, so the general method is utilizing a mechanical arm to grab fruits to reduce mechanical injuries. Most existing CNN-based picking robots are based on one fruit kind, However, the time of fruit harvest is not continuous, therefore, robots are, most, of the time idle. That generates averse economic effectiveness, because robots have high manufacturing expenses but low use ratio. According to the advantages of CNNs, they can directly extract features from input images; therefore, scholars can develop algorithms that can detect and locate a variety of fruits (Saedi and Khosravi, 2020). The mode of multitask operations can improve the use ratio of harvest robots that ensures fruit harvest robots' commercial value.

In CNN-based fruit quality grading, detection methods based on RGB images can only identify external defects, and detection methods based on hyperspectral and infrared images are focused more on internal trait detection. Results of a single detection technique are biased. Simultaneous detection of multiple quality parameters and comprehensive evaluation are a good improving trend. In addition, detection algorithms and hardware should be optimized with increasing detection difficulty.

CONCLUSION

The perishability and fragility of fruits make fruits use more labor force for careful care during the production process, which is also the reason why most fruits are expensive. At present, many researchers are bringing artificial intelligence into the field of fruit production and are carrying out a series of research studies on the use of machine vision to identify fruits. In this article, the principle of CNNs and implementation of CNN-based detection methods is elaborated, enabling researchers to better understand CNNs and their applications in fruit detection. This review emphasizes the application of CNNs in fresh fruit production, including detection of fruit flowers,

detection of fruits in the expansion period, detection of fruits in the harvest period, and detection of fruits before entering the market. We have performed a lot of investigations and analyses of literature in this area and presented in detail the convolution models, improvement points, training methods, detected objects, and final detection results in these studies. Through our investigation of experiments, we found that CNNs do have exceptional performance in the detection of fruits. However, this does not mean that fruit detection should evolve toward a single direction of detection based on CNNs. Through our comprehension and comparison of current research, we summarized the challenges that researchers encountered when using CNNs for fruit recognition and discussed future development trends.

AUTHOR CONTRIBUTIONS

CW, JX, and ZZ designed the survey. SL, BZ, LL, GL, YW, and PH collected and analyzed the data, and wrote the manuscript. JX and ZZ revised the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This research was supported by the National Natural Science Foundation of China (52005069 and 32071912) and the China Postdoctoral Science Foundation (2020M683379).

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2016). TensorFlow: large-scale machine learning on heterogeneous distributed systems. *arXiv* [Preprint]. Available online at: https://arxiv.org/abs/1603.04467 (accessed September 2021).

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., et al. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* 8:53. doi: 10.1186/s40537-021-00444-8

Ananthanarayana, T., Ptucha, R., and Kelly, S. C. (2020). Deep learning based fruit freshness classification and detection with CMOS image sensors and edge processors. *Electron. Imaging* 2020, 172-1–172-7.

Angermueller, C., Pärnamaa, T., Parts, L., and Stegle, O. (2016). Deep learning for computational biology. *Mol. Syst. Biol.* 12:878. doi: 10.15252/msb.20156651

Apolo-Apolo, O. E., Martínez-Guanter, J., Egea, G., Raja, P., and Pérez-Ruiz, M. (2020). Deep learning techniques for estimation of the yield and size of citrus fruits using a UAV. *Eur. J. Agron.* 115:126030. doi: 10.1016/j.eja.2020.126030

Aquino, A., Ponce, J. M., and Andújar, J. M. (2020). Identification of olive fruit, in intensive olive orchards, by means of its morphological structure using convolutional neural networks. *Comput. Electron. Agric.* 176:105616. doi: 10.1016/j.compag.2020.105616

Ashraf, S., Kadery, I., Chowdhury, A. A., Mahbub, T. Z., and Rahman, R. M. (2019). Fruit image classification using convolutional neural networks. *Int. J. Softw. Innov.* 7, 51–70. doi: 10.4018/IJSI.2019100103

Azizah, L. M., Umayah, S. F., Riyadi, S., Damarjati, C., and Utama, N. A. (2017). "Deep learning implementation using convolutional neural network in mangosteen surface defect detection," in *Proceedings of the 2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, Penang.

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). SegNet: a deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 2481–2495. doi: 10.1109/TPAMI.2016.2644615

Bao, J., Chen, D., Wen, F., Li, H., and Hua, G. (2017). "CVAE-GAN: fine-grained image generation through asymmetric training," in *Proceedings of the 2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE),* Penang.

Bargoti, S., and Underwood, J. P. (2017). Image segmentation for fruit detection and yield estimation in apple orchards. *J. Field Robot.* 34, 1039–1060. doi: 10.1002/rob.21699

Behera, S. K., Rath, A. K., and Sethy, P. K. (2021). Fruits yield estimation using faster R-CNN with MIoU. *Multimed. Tools Appl.* 80, 19043–19056. doi: 10.1007/s11042-021-10704-7

Bochkovskiy, A., Wang, C., and Liao, H. M. (2020). YOLOv4: optimal speed and accuracy of object detection. *arXiv* [Preprint]. Available online at: https://arxiv.org/pdf/2004.10934.pdf (accessed September 2021).

Bolya, D., Zhou, C., Xiao, F., and Lee, Y. J. (2019). "YOLACT: real-time instance segmentation," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)9156-9165*, Seoul. doi: 10.1109/ICCV.2019.00925

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

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* 16, 321–357. doi: 10.1613/jair.953

Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2018). DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 834–848. doi: 10.1109/TPAMI.2017.2699184

Chen, S. W., Shivakumar, S. S., Dcunha, S., Das, J., Okon, E., Qu, C., et al. (2017). Counting apples and oranges with deep learning: a data-driven approach. *IEEE Robot. Autom. Lett.* 2, 781–788. doi: 10.1109/LRA.2017.2651944

Chen, Y., An, X., Gao, S., Li, S., and Kang, H. (2021). A deep learning-based vision system combining detection and tracking for fast on-line citrus sorting. *Front. Plant Sci.* 12:622062. doi: 10.3389/fpls.2021.622062

Chen, Y., Lee, W. S., Gan, H., Peres, N., Fraisse, C., Zhang, Y., et al. (2019). Strawberry yield prediction based on a deep neural network using high-resolution aerial orthoimages. *Remote Sens.* 11:1584. doi: 10.3390/rs11131584

Cheng, T., Wang, X., Huang, L., and Liu, W. (2020). "Boundary-preserving mask R-CNN," in *Proceedings of the European Conference on Computer Vision*, Glasgow, 660–676. doi: 10.1007/978-3-030-58568-6_39

Chu, P., Li, Z., Lammers, K., Lu, R., and Liu, X. (2021). Deep learning-based apple detection using a suppression mask R-CNN. *Pattern Recogn. Lett.* 147, 206–211. doi: 10.1016/j.patrec.2021.04.022

Cortes, C., and Vapnik, V. (1995). Support-vector networks. *Mach. Learn.* 20, 273–297. doi: 10.1023/A:1022627411411

Couprie, C., Farabet, C., Najman, L., and LeCun, Y. (2013). *Indoor Semantic Segmentation Using Depth Information*. Available online at: https://hal.archives-ouvertes.fr/hal-00805105 (accessed September 2021).

Da Costa, A. Z., Figueroa, H. E. H., and Fracarolli, J. A. (2020). Computer vision based detection of external defects on tomatoes using deep learning. *Biosyst. Eng.* 190, 131–144. doi: 10.1016/j.biosystemseng.2019.12.003

Dashuta, A., and Klapp, I. (2018). *Melon Recognition in UAV Images to Estimate Yield of a Breeding Process*. Available online at: https://opg.optica.org/abstract.cfm?uri=EE-2018-ET4A.2 (accessed September 2021).

De Luna, R. G., Dadios, E. P., Bandala, A. A., and Vicerra, R. R. P. (2019). "Tomato fruit image dataset for deep transfer learning-based defect detection," *Proceedings of the 2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, Bangkok, 356–361.

Deng, Y., Wu, H., and Zhu, H. (2020). Recognition and counting of citrus flowers based on instance segmentation. *Nong Ye Gong Cheng Xue Bao* 36, 200–207. doi: 10.11975/j.issn.1002-6819.2020.07.023

Dias, P. A., Tabb, A., and Medeiros, H. (2018). Apple flower detection using deep convolutional networks. *Comput. Ind.* 99, 17–28. doi: 10.1016/j.compind.2018.03.010

Dong, W., Roy, P., Peng, C., and Isler, V. (2021). Ellipse R-CNN: learning to infer elliptical object from clustering and occlusion. *IEEE Trans. Image Process.* 30,2193–2206. doi: 10.1109/TIP.2021.3050673

Dong, Y., Tao, J., Zhang, Y., Lin, W., and Ai, J. (2021). Deep learning in aircraft design, dynamics, and control: review and prospects. *IEEE Trans. Aerospace Electron. Syst.* 57, 2346–2368. doi: 10.1109/TAES.2021.3056086

Fan, S., Li, J., Zhang, Y., Tian, X., Wang, Q., He, X., et al. (2020). On line detection of defective apples using computer vision system combined with deep learning methods. *J. Food Eng.* 286:110102. doi: 10.1016/j.jfoodeng.2020.110102

Farabet, C., Couprie, C., Najman, L., and LeCun, Y. (2013). Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1915–1929. doi: 10.1109/TPAMI.2012.231

Farjon, G., Krikeb, O., Hillel, A. B., and Alchanatis, V. (2020). Detection and counting of flowers on apple trees for better chemical thinning decisions. *Precis. Agric.* 21, 503–521. doi: 10.1007/s11119-019-09679-1

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognit. Lett.* 27 861–874.

Fu, L., Feng, Y., Majeed, Y., Zhang, X., Zhang, J., Karkee, M., et al. (2018). Kiwifruit detection in field images using faster R-CNN with ZFNet. *IFAC Papersonline* 51, 45–50. doi: 10.1016/j.ifacol.2018.08.059

Fu, L., Feng, Y., Wu, J., Liu, Z., Gao, F., Majeed, Y., et al. (2021). Fast and accurate detection of kiwifruit in orchard using improved YOLOv3-tiny model. *Precis. Agric.* 22, 754–776. doi: 10.1007/s11119-020-09754-y

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

Fu, L., Majeed, Y., Zhang, X., Karkee, M., and Zhang, Q. (2020b). Faster R–CNN–based apple detection in dense-foliage fruiting-wall trees using RGB and depth features for robotic harvesting. *Biosyst. Eng.* 197, 245–256. doi: 10.1016/j.biosystemseng.2020.07.007

Fukushima, K. (1980). Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* 36, 193–202. doi: 10.1007/BF00344251

Gai, R., Chen, N., and Yuan, H. (2021). A detection algorithm for cherry fruits based on the improved YOLO-v4 model. *Neural Comput. Appl.* doi: 10.1007/s00521-021-06029-z

Ganesh, P., Volle, K., Burks, T. F., and Mehta, S. S. (2019). Deep orange: Mask R-CNN based orange detection and segmentation. *IFAC Papersonline* 52, 70–75. doi: 10.1016/j.ifacol.2019.12.499

Gao, F., Fu, L., Zhang, X., Majeed, Y., Li, R., Karkee, M., et al. (2020). Multi-class fruit-on-plant detection for apple in SNAP system using faster R-CNN. *Comput. Electron. Agric.* 176:105634. doi: 10.1016/j.compag.2020.105634

Ge, Y., Xiong, Y., Tenorio, G. L., and From, P. J. (2019). Fruit localization and environment perception for strawberry harvesting robots. *IEEE Access* 7, 147642–147652. doi: 10.1109/ACCESS.2019.2946369

Gené-Mola, J., Gregorio, E., Guevara, J., Auat, F., Sanz-Cortiella, R., Escolà, A., et al. (2019a). Fruit detection in an apple orchard using a mobile terrestrial laser scanner. *Biosyst. Eng.* 187, 171–184. doi: 10.1016/j.biosystemseng.2019.08.017

Gené-Mola, J., Vilaplana, V., Rosell-Polo, J. R., Morros, J., Ruiz-Hidalgo, J., and Gregorio, E. (2019b). Multi-modal deep learning for fuji apple detection using RGB-D cameras and their radiometric capabilities. *Comput. Electron. Agric.* 162, 689–698. doi: 10.1016/j.compag.2019.05.016

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

Girshick, R. (2015). *fast r-cnn*. Available online at: https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Girshick_Fast_R-CNN_ICCV_2015_paper.pdf (accessed September 2021).

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, 580–587. doi: 10.1109/CVPR.2014.81

Glorot, X., and Bengio, Y. (2010). "Understanding the difficulty of training deep feed forward neural networks," in *Proceedings of the 13th 2010 International Conference on Artificial Intelligence and Statistics*, Vol. 9, Sardinia, 249–256.

Gongal, A., Silwal, A., Amatya, S., Karkee, M., Zhang, Q., and Lewis, K. (2016). Apple crop-load estimation with over-the-row machine vision system. *Comput. Electron. Agric.* 120, 26–35. doi: 10.1016/j.compag.2015.10.022

Gonzalez, S., Arellano, C., and Tapia, J. E. (2019). Deepblueberry: quantification of blueberries in the wild using instance segmentation. *IEEE Access* 7, 105776–105788. doi: 10.1109/ACCESS.2019.2933062

Gupta, A., Harrison, P. J., Wieslander, H., Pielawski, N., Kartasalo, K., Partel, G., et al. (2019). Deep learning in image cytometry: a review. *Cytometry A* 95, 366–380. doi: 10.1002/cyto.a.23701

Hariharan, B., Arbeláez, P., Girshick, R., and Malik, J. (2014). "Simultaneous detection and segmentation," in *Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science*, Vol. 8695, eds D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars (Cham: Springer), 297–312. doi: 10.1007/978-3-319-10584-0_20

He, K., Gkioxari, G., Dollar, P., and Girshick, R. (2017). *mask r-cnn*. Available online at: https://openaccess.thecvf.com/content_ICCV_2017/papers/He_Mask_R-CNN_ICCV_2017_paper.pdf

He, K., Zhang, X., Ren, S., and Sun, J. (2015). "Delving deep into rectifiers: surpassing human-level performance on ImageNet classification," in *Proceedings of the International Conference on Computer Vision*, Santiago, 1026–1034. doi: 10.1109/ICCV.2015.123

He, K. M., Zhang, X. Y., Ren, S. Q., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 770–778. doi: 10.1109/CVPR.2016.90

He, L., Fu, H., Karkee, M., and Zhang, Q. (2017). Effect of fruit location on apple detachment with mechanical shaking. *Biosyst. Eng.* 157, 63–71. doi: 10.1016/j.biosystemseng.2017.02.009

He, T., Zhang, Z., Zhang, H., Zhang, Z., and Li, M. (2019). "Bag of tricks for image classification with convolutional neural networks," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA.

Hinton, G. E., Osindero, S., and Teh, Y. (2006). A fast learning algorithm for deep belief nets. *Neural Comput.* 18, 1527–1554. doi: 10.1162/neco.2006.18.7.1527

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). MobileNets: efficient convolutional neural networks for mobile vision applications. *arXiv* [Preprint]. Available online at: http://arxiv.org/abs/1704.04861 (accessed September 2021).

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). "Densely connected convolutional networks," in *Proceedings of the 2017 30th IEEE conference on computer vision and pattern recognition (CVPR)*, Honolulu, HI, 2261–2269. doi: 10.1109/CVPR.2017.243

Huang, H., Huang, T., Li, Z., Lyu, S., and Hong, T. (2022). Design of citrus fruit detection system based on mobile platform and edge computer device. *Sensors* 22:59. doi: 10.3390/s22010059

Huang, Y., Wang, T., and Basanta, H. (2020). Using fuzzy mask R-CNN model to automatically identify tomato ripeness. *IEEE Access* 8, 207672–207682. doi: 10.1109/ACCESS.2020.3038184

Huang, Z., Huang, L., Gong, Y., Huang, C., and Wang, X. (2019). "Mask Scoring R-CNN," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA.

Jahanbakhshi, A., Momeny, M., Mahmoudi, M., and Zhang, Y. (2020). Classification of sour lemons based on apparent defects using stochastic pooling mechanism in deep convolutional neural networks. *Sci. Hortic.* 263:109133. doi: 10.1016/j.scienta.2019.109133

Janowski, A., Kaźmierczak, R., Kowalczyk, C., and Szulwic, J. (2021). Detecting apples in the wild: potential for harvest quantity estimation. *Sustainability* 13:8054. doi: 10.3390/su13148054

Jia, W., Tian, Y., Luo, R., Zhang, Z., Lian, J., and Zheng, Y. (2020). Detection and segmentation of overlapped fruits based on optimized mask R-CNN application in apple harvesting robot. *Comput. Electron. Agric.* 172:105380. doi: 10.1016/j.compag.2020.105380

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., et al. (2014). "Caffe: convolutional architecture for fast feature embedding," in *Proceedings of the 2014 22nd ACM international conference on Multimedia*, New York; NY.

Jiang, B., He, J., Yang, S., Fu, H., Li, T., Song, H., et al. (2019). Fusion of machine vision technology and AlexNet-CNNs deep learning network for the detection of postharvest apple pesticide residues. *Artif. Intell. Agric.* 1, 1–8. doi: 10.1016/j.aiia.2019.02.001

Jie, D., Wu, S., Wang, P., Li, Y., Ye, D., and Wei, X. (2021). Research on citrus grandis granulation determination based on hyperspectral imaging through deep learning. *Food Anal. Methods* 14, 280–289. doi: 10.1007/s12161-020-01873-6

Joe, G. G., Shaun, M. K., Lewis, M., and David, T. J. (2022). A guide to machine learning for biologists. *Nat. Rev. Mol. Cell Biol.* 23, 40–55.

Kalantar, A., Dashuta, A., Edan, Y., Dafna, A., Gur, A., and Klapp, I. (2019). "Estimating melon yield for breeding processes by machine-vision processing of UAV images," in *Proceedings of the 12th European Conference on Precision Agriculture, ECPA 2019)*, ed. J. V. Stafford (Wageningen: Wageningen Academic Publishers), 381–387. doi: 10.3920/978-90-8686-888-9_47

Kalantar, A., Edan, Y., Gur, A., and Klapp, I. (2020). A deep learning system for single and overall weight estimation of melons using unmanned aerial vehicle images. *Comput. Electron. Agric.* 178:105748. doi: 10.1016/j.compag.2020.105748

Kalogerakis, E., Averkiou, M., Maji, S., and Chaudhuri, S. (2017). "3D shape segmentation with projective convolutional networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI.

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

Kang, H., and Chen, C. (2020). Fast implementation of real-time fruit detection in apple orchards using deep learning. *Comput. Electron. Agric.* 168:105108. doi: 10.1016/j.compag.2019.105108

Khosravi, H., Saedi, S. I., and Rezaei, M. (2021). Real-time recognition of on-branch olive ripening stages by a deep convolutional neural network. *Sci. Hortic.* 287:110252. doi: 10.1016/j.scienta.2021.110252

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

Koirala, A., Walsh, K. B., Wang, Z., and McCarthy, C. (2019a). Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of 'MangoYOLO'. *Precis. Agric.* 20, 1107–1135. doi: 10.1007/s11119-019-09642-0

Krizhevsky, A., Sutskever, I., and Hinton, G. (2017). ImageNet classification with deep convolutional neural networks. *Commun. ACM* 60, 84–90. doi: 10.1145/3065386

Lamb, N., and Chuah, M. C. (2018). "A strawberry detection system using convolutional neural networks," in *Proceedings of the 2018 IEEE International Conference on Big Data(Big Data)*, Seattle, WA, 2515–2520. doi: 10.1109/BigData.2018.8622466

LeCun, Y., and Bengio, Y. (1995). "Convolutional networks for images," in *Speech, and Time-Series. Handbook of Brain Theory & Neural Networks*, ed. M. A. Arbib (Cambridge, MA: MIT Press), 1–14.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., et al. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Comput.* 1, 541–551. doi: 10.1162/neco.1989.1.4.541

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324. doi: 10.1109/9780470544976.ch9

Li, G., Huang, Y., Chen, Z., Chesser, J., Gary, D., Purswell, J. L., et al. (2021). Practices and applications of convolutional neural network-based computer vision systems in animal farming: a review. *Sensors* 21:1492. doi: 10.3390/s21041492

Lin, G., Tang, Y., Zou, X., Li, J., and Xiong, J. (2019). In-field citrus detection and localisation based on RGB-D image analysis. *Biosyst. Eng.* 186, 34–44. doi: 10.1016/j.biosystemseng.2019.06.019

Lin, G., Tang, Y., Zou, X., and Wang, C. (2021). Three-dimensional reconstruction of guava fruits and branches using instance segmentation and geometry analysis. *Comput. Electron. Agric.* 184:106107. doi: 10.1016/j.compag.2021.106107

Lin, P., Lee, W. S., Chen, Y. M., Peres, N., and Fraisse, C. (2020). A deep-level region-based visual representation architecture for detecting strawberry flowers in an outdoor field. *Precis. Agric.* 21, 387–402. doi: 10.1007/s11119-019-09673-7

Liu, S., Jia, J., Fidler, S., and Urtasun, R. (2017). "SGN: sequential grouping networks for instance segmentation," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, 3516–3524.

Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. (2018). "Path aggregation network for instance segmentation," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern*, Salt Lake City, UT.

Liu, S., Qi, X., Shi, J., Zhang, H., and Jia, J. (2016). "Mufti-scale Patch aggregation(MPA)for Simultaneous Detection and Segmentation}G]," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 3141–3149.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., et al. (2016). "SSD: single shot MultiBox detector," in *Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science*, eds B. Leibe, J. Matas, N. Sebe, and M. Welling (Cham: Springer), 21–37. doi: 10.1007/978-3-319-46448-0_2

Liu, Y. P., Yang, C., Ling, H., Mabu, S., and Kuremoto, T. (2018). "A visual system of citrus picking robot using convolutional neural networks," in *Proceedings of the 2018 5th International Conference on Systems and Informatics (ICSAI)*, Nanjing, 344–349. doi: 10.1109/ICSAI.2018.8599325

Liu, Z., Wu, J., Fu, L., Majeed, Y., Feng, Y., Li, R., et al. (2019). Improved kiwifruit detection using pre-trained VGG16 with RGB and NIR information fusion. *IEEE Access* 8, 2327–2336. doi: 10.1109/ACCESS.2019.2962513

Longye, X., Zhuo, W., Haishen, L., Xilong, K., and Changhui, Y. (2019). Overlapping citrus segmentation and reconstruction based on mask R-CNN model and concave region simplification and distance analysis. *J. Phys. Conf. Ser.* 1345:32064. doi: 10.1088/1742-6596/1345/3/032064

Majeed, Y., Zhang, J., Zhang, X., Fu, L., Karkee, M., Zhang, Q., et al. (2020). Deep learning based segmentation for automated training of apple trees on trellis wires. *Comput. Electron. Agric.* 170:105277. doi: 10.1016/j.compag.2020.105277

Mohsen, Y. N., Dan, T., and Milad, E. (2021). Using hybrid artificial intelligence and evolutionary optimization algorithms for estimating soybean yield and fresh biomass using hyperspectral vegetation indices. *Remote Sens.* 13:2555. doi: 10.3390/rs13132555

Momeny, M., Jahanbakhshi, A., Jafarnezhad, K., and Zhang, Y. (2020). Accurate classification of cherry fruit using deep CNN based on hybrid pooling approach. *Postharvest Biol. Technol.* 166:111204. doi: 10.1016/j.postharvbio.2020.111204

Mu, L., Gao, Z., Cui, Y., Li, K., Liu, H., and Fu, L. (2019). Kiwifruit detection of far-view and occluded fruit based on improved AlexNet. *Trans. Chin. Soc. Agric. Mach.* 50, 24–34. doi: 10.6041/j.issn.1000-1298.2019.10.003

Mu, Y., Chen, T., Ninomiya, S., and Guo, W. (2020). Intact detection of highly occluded immature tomatoes on plants using deep learning techniques. *Sensors* 20:2984. doi: 10.3390/s20102984

Naranjo-Torres, J., Mora, M., Hernández-García, R., Barrientos, R. J., Fredes, C., and Valenzuela, A. (2020). A review of convolutional neural network applied to fruit image processing. *Appl. Sci.* 10:3443. doi: 10.3390/app10103443

Nguyen, H., Kieu, L., Wen, T., and Cai, C. (2018). Deep learning methods in transportation domain: a review. *IET Intell. Transp. Syst.* 12, 998–1004. doi: 10.1049/iet-its.2018.0064

Nguyen, T. T., Vandevoorde, K., Wouters, N., Kayacan, E., De Baerdemaeker, J. G., and Saeys, W. (2016). Detection of red and bicoloured apples on tree with an RGB-D camera. *Biosyst. Eng.* 146, 33–44. doi: 10.1016/j.biosystemseng.2016.01.007

Ni, X., Li, C., Jiang, H., and Takeda, F. (2020). Deep learning image segmentation and extraction of blueberry fruit traits associated with harvestability and yield. *Hortic. Res.* 7:110. doi: 10.1038/s41438-020-0323-3

Ni, X., Li, C., Jiang, H., and Takeda, F. (2021). Three-dimensional photogrammetry with deep learning instance segmentation to extract berry fruit harvestability traits. *ISPRS J. Photogramm. Remote Sens.* 171, 297–309. doi: 10.1016/j.isprsjprs.2020.11.010

Nur Alam, M., Saugat, S., Santosh, D., Sarkar, M. I., and Al-Absi, A. A. (2020). "Apple defect detection Q64 based on deep convolutional neural network," in *Proceedings of the 2020 International Conference on Smart Computing and Cyber Security: Strategic Foresight, Security Challenges and Innovation*, ed. P. K. Pattnaik (Singapore: Springer Verlag), 215–223. doi: 10.1007/978-981-15-7990-5_21

Nyarko, E. K., Vidović, I., Radočaj, K., and Cupec, R. (2018). A nearest neighbor approach for fruit recognition in RGB-D images based on detection of convex surfaces. *Expert Syst. Appl.* 114, 454–466. doi: 10.1016/j.eswa.2018.07.048

Okamoto, H., and Lee, W. S. (2009). Green citrus detection using hyperspectral imaging. *Comput. Electron. Agric.* 66, 201–208. doi: 10.1016/j.compag.2009.02.004

Onishi, Y., Yoshida, T., Kurita, H., Fukao, T., Arihara, H., and Iwai, A. (2019). An automated fruit harvesting robot by using deep learning. *ROBOMECH J.* 6, 1–8. doi: 10.1186/s40648-019-0141-2

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2014). "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the 2014 Computer Vision & Pattern Recognition*, Columbus, OH.

Pal, N. R., and Pal, S. K. (1993). A review on image segmentation techniques. *Pattern Recogn.* 26, 1277–1294. doi: 10.1016/0031-3203(93)90135-J

Palacios, F., Bueno, G., Salido, J., Diago, M. P., Hernández, I., and Tardaguila, J. (2020). Automated grapevine flower detection and quantification method based on computer vision and deep learning from on-the-go imaging using a mobile sensing platform under field conditions. *Comput. Electron. Agric.* 178:105796. doi: 10.1016/j.compag.2020.105796

Parvathi, S., and Tamil Selvi, S. (2021). Detection of maturity stages of coconuts in complex background using faster R-CNN model. *Biosyst. Eng.* 202, 119–132. doi: 10.1016/j.biosystemseng.2020.12.002

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). "PyTorch: an imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, eds H. Wallach, H. Larochelle, A. Beygelzimer, F. D. Alché'-Buc, E. Fox, and R. Garnett (New York, NY: Curran Associates, Inc), 8024–8035.

Patil, P. U., Lande, S. B., Nagalkar, V. J., Nikam, S. B., and Wakchaure, G. C. (2021). Grading and sorting technique of dragon fruits using machine learning algorithms. *J. Agric. Food Res.* 4:100118. doi: 10.1016/j.jafr.2021.100118

Peng, H., Huang, B., Shao, Y., Li, Z., Zhang, Z., Chen, Y., et al. (2018). General improved SSD model for picking object recognition of multiple fruits in natural environment. *Nong Ye Gong Cheng Xue Bao* 34, 155–162. doi: 10.11975/j.issn.1002-6819.2018.16.020

Pérez-Borrero, I., Marín-Santos, D., Gegúndez-Arias, M. E., and Cortés-Ancos, E. (2020). A fast and accurate deep learning method for strawberry instance segmentation. *Comput. Electron. Agric.* 178:105736. doi: 10.1016/j.compag.2020.105736

Pérez-Borrero, I., Marin-Santos, D., Vasallo-Vazquez, M. J., and Gegundez-Arias, M. E. (2021). A new deep-learning strawberry instance segmentation methodology based on a fully convolutional neural network. *Neural Comput. Appl.* 33, 15059–15071.

Pinheiro, P. O., Collobert, R., and Dollar, P. (2015). Learning to segment object candidates. *arXiv* [Preprint]. Available online at: https://arxiv.org/pdf/1506.06204.pdf (accessed September 2021).

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017a). "PointNet: deep learning on point Sets for 3D classification and segmentation," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017b). *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*. (Cambridge, MA: MIT Press), 5099–5108.

Quinlan, J. R. (1986). Induction of decision trees. *Mach. Learn.* 1, 81–106.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 779–788.

Redmon, J., and Farhadi, A. (2017). "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, 6517–6525.

Redmon, J., and Farhadi, A. (2018). YOLOv3: an incremental improvement. *arXiv* [Preprint]. Available online at: http://arxiv.org/abs/1804.02767v1 (accessed September 2021).

Rehman, H. U., and Miura, J. (2021). Viewpoint planning for automated fruit harvesting using deep learning," in *Proceedings of the 2021 IEEE/SICE International Symposium on System Integration (SII),* Iwaki, 409–414. doi: 10.1109/IEEECONF49454.2021.9382628

Rehman, S. U., Tu, S., Waqas, M., Huang, Y., Rehman, O. U., Ahmad, B., et al. (2019). Unsupervised pre-trained filter learning approach for efficient convolution neural network. *Neurocomputing* 365, 171–190. doi: 10.1016/j.neucom.2019.06.084

Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Riegler, G., Ulusoy, A. O., and Geiger, A. (2016). "Octnet: learning deep 3d representations at high resolutions," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition* (Piscataway, NJ: IEEE).

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-net: convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science*, Vol. 9351, eds N. Navab, J. Hornegger, W. Wells and A. Frangi (Cham: Springer), 234–241. doi: 10.1007/978-3-319-24574-4_28

Rudolph, R., Herzog, K., Töpfer, R., and Steinhage, V. (2019). Efficient identification, localization and quantification of grapevine inflorescences and flowers in unprepared field images using fully convolutional networks. *Vitis* 58, 95–104. doi: 10.5073/vitis.2019.58.95-104

Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. (2007). LabelMe: a database and web-based tool for image annotation. *Int. J. Comput. Vis.* 77, 157–173. doi: 10.1007/s11263-007-0090-8

Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., and McCool, C. (2016). Deepfruits: a fruit detection system using deep neural networks. *Sensors* 16:1222. doi: 10.3390/s16081222

Saedi, S. I., and Khosravi, H. (2020). A deep neural network approach towards real-time on-branch fruit recognition for precision horticulture. *Expert Syst. Appl.* 159:113594. doi: 10.1016/j.eswa.2020.113594

Santos, T. T., de Souza, L. L., dos Santos, A. A., and Avila, S. (2020). Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association. *Comput. Electron. Agric.* 170:105247. doi: 10.1016/j.compag.2020.105247

Schuster, M., and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* 45, 2673–2681.

Shelhamer, E., Long, J., and Darrell, T. (2017). Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 640–651. doi: 10.1109/TPAMI.2016.2572683

Silwal, A., Karkee, M., and Zhang, Q. (2016). A hierarchical approach to apple identification for robotic harvesting. *Trans. ASABE* 59, 1079–1086. doi: 10.13031/trans.59.11619

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *Comput. Sci.*

Song, Z., Fu, L., Wu, J., Liu, Z., Li, R., and Cui, Y. (2019). Kiwifruit detection in field images using faster R-CNN with VGG16. *IFAC Papersonline* 52, 76–81. doi: 10.1016/j.ifacol.2019.12.500

Stein, M., Bargoti, S., and Underwood, J. (2016). Image based mango fruit detection, localisation and yield estimation using multiple view geometry. *Sensors* 16:1915. doi: 10.3390/s16111915

Sun, J., He, X., Ge, X., Wu, X., Shen, J., and Song, Y. (2018). Detection of key organs in tomato based on deep migration learning in a complex background. *Agriculture* 8:196. doi: 10.3390/agriculture8120196

Sun, J., He, X., Wu, M., Wu, X., Shen, J., and Lu, B. (2020). Detection of tomato organs based on convolutional neural network under the overlap and occlusion backgrounds. *Mach. Vis. Appl.* 31:31. doi: 10.1007/s00138-020-01081-6

Supper, J., Spieth, C., and Zell, A. (2007). "Reconstructing linear gene regulatory networks," in *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, eds E. Marchiori, J. H. Moore, and J. C. Rajapakse (Berlin: Springer), 270–279. doi: 10.1007/978-3-540-71783-6_26

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., and Rabinovich, A. (2015). "Going deeper with convolutions," in *Proceedings of the 2015 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 1–9. doi: 10.1109/CVPR.2015.7298594

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). "Rethinking the inception architecture for computer vision," in *Proceedings of the 2016 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2818–2826. doi: 10.1109/CVPR.2016.308

Tan, K., Lee, W. S., Gan, H., and Wang, S. (2018). Recognising blueberry fruit of different maturity using histogram oriented gradients and colour features

in outdoor scenes. *Biosyst. Eng.* 176, 59–72. doi: 10.1016/j.biosystemseng.2018.08.011

Tang, C., Chen, H., Li, X., Li, J., Zhang, Z., and Hu, X. (2021). *Look Closer to Segment Better: Boundary Patch Refinement for Instance Segmentation*. Available online at: https://docplayer.net/amp/218770859-Look-closer-to-segment-better-boundary-patch-refinement-for-instance-segmentation-supplementary-material.html (accessed September 2021).

Thendral, R., Suhasini, A., and Senthil, N. (2014). "A comparative analysis of edge and color based segmentation for orange fruit recognition," in *Proceedings of the 2014 International Conference on Communication and Signal Processing (ICCSP)*, Melmaruvathur. doi: 10.1109/ICCSP.2014.6949884

Tian, Y., Yang, G., Wang, Z., Li, E., and Liang, Z. (2020). Instance segmentation of apple flowers using the improved mask R-CNN model. *Biosyst. Eng.* 193, 264–278.

Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., and Liang, Z. (2019). Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Comput. Electron. Agric.* 157, 417–426. doi: 10.1016/j.compag.2019.01.012

Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. (2002). "Bundle adjustment — A modern synthesis," in *Vision Algorithms: Theory and Practice. IWVA 1999. Lecture Notes in Computer Science*, Vol. 1883, eds B. Triggs, A. Zisserman, R. Szeliski (Berlin: Springer), 298–372. doi: 10.1007/3-540-44480-7_21

Tsai, C., and Chiu, C. (2008). Developing a feature weight self-adjustment mechanism for a K-means clustering algorithm. *Comput. Stat. Data Anal.* 52, 4658–4672. doi: 10.1016/j.csda.2008.03.002

Tsoulias, N., Paraforos, D. S., Xanthopoulos, G., and Zude-Sasse, M. (2020). Apple shape detection based on geometric and radiometric features using a LiDAR laser scanner. *Remote Sens.* 12:2481. doi: 10.3390/rs12152481

Tu, S., Pang, J., Liu, H., Zhuang, N., Chen, Y., Zheng, C., et al. (2020). Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images. *Precis. Agric.* 21, 1072–1091. doi: 10.1007/s11119-020-09709-3

Tu, S., Xue, Y., Zheng, C., Qi, Y., Wan, H., and Mao, L. (2018). Detection of passion fruits and maturity classification using red-green-blue depth images. *Biosyst. Eng.* 175, 156–167. doi: 10.1016/j.biosystemseng.2018.09.004

Uijlings, J. R., van de Sande, K. E. A., Gevers, T., and Smeulders, A. W. M. (2013). Selective search for object recognition. *Int. J. Comput. Vis.* 104, 154–171. doi: 10.1007/s11263-013-0620-5

Vasconez, J. P., Delpiano, J., Vougioukas, S., and AuatCheein, F. (2020). Comparison of convolutional neural networks in fruit detection and counting: a comprehensive evaluation. *Comput. Electron. Agric.* 173:105348. doi: 10.1016/j.compag.2020.105348

Wan, S., and Goudos, S. (2020). Faster R-CNN for multi-class fruit detection using a robotic vision system. *Comput. Netw.* 168:107036. doi: 10.1016/j.comnet.2019.107036

Wang, H., Dong, L., Zhou, H., Luo, L., Lin, G., Wu, J., et al. (2021). YOLOv3-litchi detection method of densely distributed litchi in large vision scenes. *Math. Probl. Eng.* 2021:8883015. doi: 10.1155/2021/8883015

Wang, H., Li, X., Li, Y., Sun, Y., and Xu, H. (2020). Non-destructive detection of apple multi-quality parameters based on hyperspectral imaging technology and 3D-CNN. *Nanjing NongyeDaxueXuebao* 43, 178–185. doi: 10.7685/jnau.201906067

Wang, J., Chen, Y., Zeng, Z., Li, J., Liu, W., and Zou, X. (2018). Extraction of litchi fruit pericarp defect based on a fully convolutional neural network. *Hua Nan Nong Ye Da XueXue Bao* 39, 104–110. doi: 10.7671/j.issn.1001-411X.2018.06.016

Wang, X., Tang, J., and Whitty, M. (2021). DeepPhenology: estimation of apple flower phenology distributions based on deep learning. *Comput. Electron. Agric.* 185:106123. doi: 10.1016/j.compag.2021.106123

Wei, C., Han, W., and Liu, H. (2021). Counting method of cherry tomato fruits in greenhouses based on deep learning. *J. China Univ. Metrol.* 32, 93–100. doi: 10.3969/j.issn.2096-2835.2021.01.013

Wittstruck, L., Kühling, I., Trautz, D., Kohlbrecher, M., and Jarmer, T. (2021). UAV-based RGB imagery for Hokkaido pumpkin (*Cucurbita max.*) detection and yield estimation. *Sensors* 21:118. doi: 10.3390/s21010118

Wouters, N., De Ketelaere, B., De Baerdemaeker, J., and Saeys, W. (2012). Hyperspectral waveband selection for automatic detection of floral pear buds. *Precis. Agric.* 14, 86–98. doi: 10.1007/s11119-012-9279-0

Wu, A., Zhu, J., and Ren, T. (2020). Detection of apple defect using laser-induced light backscattering imaging and convolutional neural network. *Comput. Electr. Eng.* 81:106454. doi: 10.1016/j.compeleceng.2019.106454

Wu, D., Lv, S., Jiang, M., and Song, H. (2020). Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. *Comput. Electron. Agric.* 178:105742. doi: 10.1016/j.compag.2020.105742

Wu, S., Tung, H., and Hsu, Y. (2020). "Deep learning for automatic quality grading of mangoes: methods and insights," in *Proceedings of the 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Miami, FL, 446–453. doi: 10.1109/ICMLA51294.2020.00076

Xiong, J., Liu, Z., Lin, R., Chen, S., Chen, W., and Yang, Z. (2018). Unmanned aerial vehicle vision detection technology of green mango on tree in natural environment. *Trans. Chin. Soc. Agric. Mach.* 49, 23–29. doi: 10.6041/j.issn.1000-1298.2018.11.003

Xiong, L., Wang, Z., Liao, H., Kang, X., and Yang, C. (2019). Overlapping citrus segmentation and reconstruction based on mask R-CNN model and concave region simplification and distance analysis. *J. Phys. Conf. Ser.* 1345:32064. doi: 10.1088/1742-6596/1345/3/032064

Xiong, J., Liu, B., Zhong, Z., Chen, S., and Zheng, Z. (2021). Litchi flower and leaf segmentation and recognition based on deep semantic segmentation. *Trans. Chin. Soc. Agric. Machinery* 52, 252–258.

Xiong, Y., Ge, Y., and From, P. J. (2020). An obstacle separation method for robotic picking of fruits in clusters. *Comput. Electron. Agric.* 175:105397. doi: 10.1016/j.compag.2020.105397

Xu, L., Huang, H., and Ding, W. (2021). Detection of small fruit target based on improved DenseNet. *J. Zhejiang Univ.* 55, 377–385. doi: 10.3785/j.issn.1008-973X.2021.02.018

Xu, S., Liu, Y., Hu, W., Wu, Y., Liu, S., Wang, Y., et al. (2020). Nondestructive detection of yellow peach quality parameters based on 3D-CNN and hyperspectral images. *J. Phys. Conf. Ser.* 1682:012030. doi: 10.1088/1742-6596/1682/1/012030

Yan, J., Zhao, Y., Zhang, L., Su, X., Liu, H., Zhang, F., et al. (2019). Recognition of *Rosa roxbunghii* in natural environment based on improved faster RCNN. *Nong Ye Gong Cheng Xue Bao* 35, 143–150. doi: 10.11975/j.issn.1002-6819.2019.18.018

Yang, C., Wang, Z., Xiong, L., Kang, X., and Zhao, W. (2019). Identification and reconstruction of citrus branches under complex background based on mask R-CNN. *Trans. Chin. Soc. Agric. Machinery* 50, 22–69. doi: 10.6041/j.issn.1000-1298.2019.08.003

Yang, C. H., Xiong, L. Y., Wang, Z., Wang, Y., Shi, G., Kuremot, T., et al. (2020). Integrated detection of citrus fruits and branches using a convolutional neural network. *Comput. Electron. Agric.* 174:105469. doi: 10.1016/j.compag.2020.105469

Yin, W., Wen, H., Ning, Z., Ye, J., Dong, Z., and Luo, L. (2021). Fruit detection and pose estimation for grape Cluster–Harvesting robot using binocular imagery based on deep neural networks. *Front. Robot. AI* 8:626989. doi: 10.3389/frobt.2021.626989

Yu, G., Ma, B., Chen, J., Li, X., Li, Y., and Li, C. (2021). Nondestructive identification of pesticide residues on the hami melon surface using deep feature fusion by Vis/NIR spectroscopy and 1D-CNN. *J. Food Process Eng.* 44:e13602. doi: 10.1111/jfpe.13602

Yu, X., Lu, H., and Wu, D. (2018). Development of deep learning method for predicting firmness and soluble solid content of postharvest korla fragrant pear using Vis/NIR hyperspectral reflectance imaging. *Postharvest Biol. Technol.* 141, 39–49. doi: 10.1016/j.postharvbio.2018.02.013

Yu, Y., Zhang, K., Yang, L., and Zhang, D. (2019). Fruit detection for strawberry harvesting robot in non-structural environment based on mask-RCNN. *Comput. Electron. Agric.* 163:104846. doi: 10.1016/j.compag.2019.06.001

Yuan, B., Zhan, J., and Chen, C. (2017). Evolution of a development model for fruit industry against background of an aging population: intensive or extensive adjustment. *Sustainability* 10:49. doi: 10.3390/su10010049

Yue, X.-Q., Shang, Z.-Y., Yang, J.-Y., Huang, L., and Wang, Y.-Q. (2020). A smart data-driven rapid method to recognize the strawberry maturity. *Inf. Process. Agric.* 7,575–584. doi: 10.1016/j.inpa.2019.10.005

Zabawa, L., Kicherer, A., Klingbeil, L., Töpfer, R., Kuhlmann, H., and Roscher, R. (2020). Counting of grapevine berries in images via semantic segmentation using convolutional neural networks. *ISPRS J. Photogramm. Remote Sens.* 164, 73–83. doi: 10.1016/j.isprsjprs.2020.04.002

Zeng, T., Wu, J., Ma, B., Wang, C., Luo, X., and Wang, W. (2019). Localization and defect detection of jujubes based on search of shortest path between frames and ensemble-CNN model. *Trans. Chin. Soc. Agric. Machinery* 50, 307–314. doi: 10.6041/j.issn.1000-1298.2019.02.035

Zhang, J., He, L., Karkee, M., Zhang, Q., Zhang, X., and Gao, Z. (2018). Branch detection for apple trees trained in fruiting wall architecture using depth features and regions-convolutional neural network (R-CNN). *Comput. Electron. Agric.* 155, 386–393. doi: 10.1016/j.compag.2018.10.029

Zhang, J., Karkee, M., Zhang, Q., Zhang, X., Yaqoob, M., Fu, L., et al. (2020). Multi-class object detection using faster R-CNN and estimation of shaking locations for automated shake-and-catch apple harvesting. *Comput. Electron. Agric.* 173:105384. doi: 10.1016/j.compag.2020.105384

Zhang, L., Jia, J., Gui, G., Hao, X., Gao, W., and Wang, M. (2018). Deep learning based improved classification system for designing tomato harvesting robot. *IEEE Access* 6, 67940–67950. doi: 10.1109/ACCESS.2018.2879324

Zhang, Q., and Gao, G. (2020). Prioritizing robotic grasping of stacked fruit clusters based on stalk location in RGB-D images. *Comput. Electron. Agric.* 172:105359. doi: 10.1016/j.compag.2020.105359

Zhao, Q., Kong, P., Min, J., Zhou, Y., Liang, Z., Chen, S., et al. (2019). A review of deep learning methods for the detection and classification of pulmonary nodules. *Sheng Wu Yi Xue Gong Cheng Xue Za Zhi* 36, 1060–1068. doi: 10.7507/1001-5515.201903027

Zhao, Y., Gong, L., Zhou, B., Huang, Y., and Liu, C. (2016). Detecting tomatoes in greenhouse scenes by combining AdaBoost classifier and colour analysis. *Biosyst. Eng.* 148, 127–137. doi: 10.1016/j.biosystemseng.2016.05.001

Zhao, Z., Zheng, P., Xu, S., and Wu, X. (2019). Object detection with deep learning: a review. *IEEE Trans. Neural Netw. Learn. Syst.* 30, 3212–3232. doi: 10.1109/TNNLS.2018.2876865

Zheng, Z., Zheng, L., and Yang, Y. (2017). "Unlabeled samples generated by GAN improve the person re-identification baseline in vitro," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV2017)*, Venice, 3774–3782. doi: 10.1109/ICCV.2017.405

Zhou, R., Damerow, L., Sun, Y., and Blanke, M. M. (2012). Using colour features of cv. 'Gala' apple fruits in an orchard in image processing to predict yield. *Precis. Agric.* 13, 568–580. doi: 10.1007/s11119-012-9269-2

Zhou, X., Lee, W. S., Ampatzidis, Y., Chen, Y., Peres, N., and Fraisse, C. (2021). Strawberry maturity classification from UAV and near-ground imaging using deep learning. *Smart Agric. Technol.* 1:100001. doi: 10.1016/j.atech.2021.100001

Zhou, Z., Song, Z., Fu, L., Gao, F., Li, R., and Cui, Y. (2020). Real-time kiwifruit detection in orchard using deep learning on android^{TM} smartphones for yield estimation. *Comput. Electron. Agric.* 179:105856. doi: 10.1016/j.compag.2020.105856

Zhu, J., Park, T., Isola, P., and Efros, A. A. (2017). "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, 2242–2251. doi: 10.1109/ICCV.2017.244

Zhu, J., Sharma, A. S., Xu, J., Xu, Y., Jiao, T., Ouyang, Q., et al. (2021). Rapid on-site identification of pesticide residues in tea by one-dimensional convolutional neural network coupled with surface-enhanced Raman scattering. *Spectrochim. Acta A Mol. Biomol. Spectrosc.* 246:118994. doi: 10.1016/j.saa.2020.118994

Zhu, N., Liu, X., Liu, Z., Hu, K., Wang, Y., Tan, J., et al. (2018). Deep learning for smart agriculture: concepts, tools, applications, and opportunities. *Int. J. Agric. Biol. Eng.* 11, 32–44. doi: 10.25165/ijabe.v11i4.4475

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.