Zhang et al. *Horticulture Research* (2021)8:119
https://doi.org/10.1038/s41438-021-00553-8

Horticulture Research
www.nature.com/hortres

ARTICLE Open Access

Easy domain adaptation method for filling the species gap in deep learning-based fruit detection

Wenli Zhang[1], Kaizhen Chen[1], Jiaqi Wang[1], Yun Shi[2] and Wei Guo[3]

Abstract

Fruit detection and counting are essential tasks for horticulture research. With computer vision technology development, fruit detection techniques based on deep learning have been widely used in modern orchards. However, most deep learning-based fruit detection models are generated based on fully supervised approaches, which means a model trained with one domain species may not be transferred to another. There is always a need to recreate and label the relevant training dataset, but such a procedure is time-consuming and labor-intensive. This paper proposed a domain adaptation method that can transfer an existing model trained from one domain to a new domain without extra manual labeling. The method includes three main steps: transform the source fruit image (with labeled information) into the target fruit image (without labeled information) through the CycleGAN network; Automatically label the target fruit image by a pseudo-label process; Improve the labeling accuracy by a pseudo-label self-learning approach. Use a labeled orange image dataset as the source domain, unlabeled apple and tomato image dataset as the target domain, the performance of the proposed method from the perspective of fruit detection has been evaluated. Without manual labeling for target domain image, the mean average precision reached 87.5% for apple detection and 76.9% for tomato detection, which shows that the proposed method can potentially fill the species gap in deep learning-based fruit detection.

Introduction

There is a vital need in the horticulture research field to understand fruit-related phenotypic traits, such as fruit number, size, and color. With the rapid development of modern computer technology, the demand for visual detection techniques in agriculture has increased. An object detection technique can obtain the location and category information of the fruit in the image, such as fruit positioning[1,2], fruit estimation[3,4], and automatic fruit picking[5,6], which is the technical basis for intelligent work in the orchard.

Recently, owing to the advantages of deep learning-based object detection techniques[7–13], which perform high detection accuracy and good model robustness, they have gradually replaced traditional detection methods and are widely applied in orchard fruit detection. On the other hand, most deep learning-based fruit detection techniques adopt the supervised learning strategy, which requires a large number of labeled fruit image datasets to train the model. However, a model generated with a dataset collected for one species may not work for another species; hence, new species always require labeling new data to train the new model, which is labor-intensive and time-consuming. Therefore, reducing the dataset labeling workload has become a topic of intense interest[14].

In the current stage, most related works use a strongly supervised labeling method[15] that requires drawing bounding boxes around the target objects with location and category information for model training. Mu et al.[16] collected fruit images of tomatoes in a greenhouse, Wang et al.[4] collected mango fruit images at night orchards, then labeled each visible target fruit in the images by tight

Correspondence: Wenli Zhang (zhangwenli@bjut.edu.cn) or Wei Guo (guowei@g.ecc.u-tokyo.ac.jp)
[1]Information Department, Beijing University of Technology, Beijing 100022, China
[2]Institute of Agricultural Resources and Regional Planning, Chinese Academy of Agricultural Sciences, Beijing 100081, China
Full list of author information is available at the end of the article

bounding boxes manually. Although the strongly supervised labeling method provides better detection performance, the labeling cost were high and time-consuming.

Some works then tried to train detection models based on weakly-supervised labeling methods to reduce the labeling cost. For example, researchers used image-level labels[17–20] (providing information on the category of objects in the image, no specific location information) and dot labels[21] (marking object location information with dots) to reduce the overall cost and time consumption by lessening the labeling time of individual labels. Bellocchio et al.[22,23] proposed a weakly supervised deep architecture that relies only on an image level binary classifier(whether the image contains instances of the fruit or not) to train the fruit counting model on source images. The unsupervised transformation learning and pseudo-label process are further combined to generate target fruit images and related labels and then applied to the fruit counting task on target images. Because pseudo labels are acquired only for the generated fruit images and different from actual target fruit images, the model did not fit well with the actual target fruit images. Lu et al.[24] used dot annotated method to perform maize tassel counting task in localized regions of the farmland. Ghosal et al.[25] proposed active learning inspired weakly supervised deep learning framework, and Lagandula et al.[26] combined dot-annotated methods with active learning methods[27,28] to reduce labeling time cost more than 50% on sorghum and wheat images. However, the weakly supervised labeling method still requires a certain amount of manual data labeling work.

Some researchers also suggested that unsupervised learning methods[29–31] can be applied to agriculture since they do not require data labeling. Wachs et al.[29] proposed a method based on K-means clustering to achieve the unsupervised detection of green apples in infrared and RGB images with an accuracy of 53.2%. Dubey et al.[30] utilized the K-means clustering algorithm to perform fruit segmentation and localization based on color features. Zhang et al.[31] proposed an unsupervised learning conditional random field image segmentation algorithm to segment plant organs such as fruits, leaves, and stems from green house plant images without manual labeling. However, in most agricultural field work, because of the complexity of the context and the diversity of objectives in the actual scenario, unsupervised learning methods did not performed as accurate as supervised learning methods. To address the high dataset labeling cost, some researchers also suggest that public available datasets[32–37] can be used to train fruit detection models. Sa et al.[32] presented the DeepFruit dataset, which contains apple, avocado, capsicum, mango, orange, rockmelon, and strawberry; Bargoti et al.[33] presented a acfr-multifruit-2016 dataset that contains mango, almond, and apple; Muresan et al.[34] presented Fruit-360 dataset that contains 131 categories of fruit images with a single background. However, owing to the different image acquisition conditions in each fruit dataset, including lighting conditions, occluding conditions, and shooting distance, the trained fruit detection model showed low generalization ability when applied to real applications, and it is also kown that train a model based on target scenes will always performs best.

Therefore, we consider to train several locally good models for each domain based on their own data for fruit detection tasks. Then the main problem shifts to how to generate labeled data for new domain efficiently, which the Generative Adversarial Networks (GAN)[38] seems to be a powerful tool for it. GAN have been widely used for image transformation tasks. Stein et al.[39] and Zhang et al.[40] proposed a GAN-based image transformation method to implement image transformation between simulated and real images for cross-domain segmentation tasks. Roy et al.[41] proposed Semantic-Aware GAN, which introduces multiple loss functions to optimize model training and can be applied to image transformation between image domains with large geometric shape differences. Valerio et al.[42] proposed to combine multiple regression leaf counting model and adversarial network idea to achieved cross-domain leaf counting for in the unlabeled target domain by extracting domain invariant features from different plant species. However, the above research mainly focuses on improving the generated image quality for image transformation, not labeling images for the new target domain. So in this paper, we propose a new method to use GAN to automatically label different fruit image datasets by only using a set of existing labeled fruit images.

The proposed method first uses the CycleGAN[43] network to transfer the source domain fruit dataset (with labeled information) to the target domain fruit dataset (without labeled information), then applies the pseudo-label method to label the target fruit dataset. Finally, it uses a self-learning method of pseudo labels further to improve the labeling accuracy. The performance of the proposed method from the perspective of fruit detection has been evaluated then by a labeled orange image dataset and unlabeled apple and tomato image dataset.

Materials and methods

Dataset acquisition

The experiments in this paper contain two datasets: CycleGAN datasets and object detection datasets.

CycleGAN datasets

The image transformation experiments used the apple2orange dataset[43] and the orange2tomato dataset.

(1)The apple2orange dataset contains orange and apple to train the image transformation model between orange and apple. The training set containes 995 apple images and 1019 orange images, while the test set containes 266 apple images and 248 orange images, with a uniform image resolution of 256 × 256 pixels.

(2)The orange2tomato dataset contains orange images from apple2orange dataset and the tomato images collected from the Internet. The training set contains 654 tomato images and 1019 orange images, while the test set contains 102 tomato images and 248 orange images, with a uniform image resolution of 256 × 256 pixels.

Object detection datasets

The following source fruit dataset and target fruit dataset were used in the fruit detection experiments:

(1) **Source orange dataset:** The dataset was collected from an orange orchard in Sichuan Province, China. In total, 664 orange images were collected using a DJI Osmo Action camera (Shenzhen DJI Science & Technology Co., Ltd.), including down-light, back-light, dense target, blocking target, and other fruit scenes. Relevant annotation tools were exploited to obtain the coordinate information of each orange annotation box, i.e., the x and y coordinates of the two points in the upper left and lower right corners of the annotation box. Afterward, the images were resized to 416 × 416, and randomly divided into a training set and a test set according to a 7:3 ratio.

(2) **Target dataset: apple and tomato dataset:**

Target apple dataset: The dataset is based on the MineApple dataset[37], which contains images of red and green apples in a variety of highly cluttered environments, with an average target fruit size of 40 × 40 pixels. In total, 504 images of red apples from the original training set were selected as the experimental training set, with an image resolution of 1280 × 720 and no data labeling. In total, 82 red apple images from the original test set were selected as the experimental test set. The images were cropped to 719 × 898 to remove the influence of fallen apples on the ground and then been labeled with relevant labeling tools for later experimental validation.

Target tomato dataset

The dataset is based on the dataset published by Mu et al.[16], which ware collected from two farms in Tokyo, Japan. The collected tomato images were pre-processed and the image resolution was set to 1920 × 1080, where the training set consisted of 598 unlabeled tomato images and the test set consisted of 150 labeled tomato images.

Among them, the orange images and apple images were collected outdoors, and the tomato images were collected indoors. Besides, most of the tomato images includes green tomato fruits, so the color features are similar to the background leaves. The differences in these collection environments, locations and shooting distances bring significant challenges to this study.

Workflow of the proposed method

In this paper, a data labeling conversion method between different species of fruits is proposed to realize the automatic data labeling of unlabeled fruit datasets and save the dataset labeling cost in detection tasks. The flowchart of the algorithm is depicted in Fig. 1.

The application context comprises a labeled source fruit dataset D_s and an unlabeled target fruit dataset D_T^U, both from Object detection dataset. We assume the sets $D_s = \{(I_S^1, l_S^1), (I_S^2, l_S^2), \ldots \ldots, (I_S^{N_S}, l_S^{N_S})\}$ and $D_T^U = \{I_T^1, I_T^2, \ldots \ldots, I_T^{N_T}\}$, where I_S and I_T represent the image in the source fruit dataset and the target fruit dataset, respectively. l_S represents the labeling information of the corresponding images in the source fruit dataset, and N represents the number of images in the dataset. The overall steps of the method are as follows:

Step 1: The fruit images were imported from the dataset D_s into the CycleGAN testing network for image transformation (the CycleGAN network is noted as M_1 and the associated model weight parameter is noted as w_1); there upon, construct a fake apple dataset D_F with the labeling information of the source fruit dataset D_s, where $D_F = \{(I_F^1, l_S^1), (I_F^2, l_S^2), \ldots \ldots, (I_F^{N_S}, l_S^{N_S})\}$ and I_F represents the transformed fake target fruit image.

Step 2: Feed dataset D_F into the fruit detection model called Improved-Yolov3[37] for training, the obtained fruit detection model is noted as M_2, and the weight parameter of the model is noted as w_2.

Step 3: Using the dataset D_T^U as the test set input model M_2, obtain the detection box of the real target fruit in the image I_T, and treat the detection box as the pseudo-label information of the image I_T. Subsequently, use the self-learning method of the pseudo label to improve the accuracy of the labels. Finally, obtain the dataset D_T^U with pseudo labels and note as D_T^L, where $D_T^L = \{(I_T^1, l_T^1), (I_T^2, l_T^2), \ldots \ldots, (I_T^{N_T}, l_T^{N_T})\}$ and l_T represent the labeling information for the associated image I_T.

Step 4: Output the above dataset D_T^L with label information.

The data labeling conversion algorithm includes the implementation of the following four functional modules.

A: Image transformation

The generative adversarial network[38] has been one of the most popular models in recent years. The model mainly improves the performance of the discriminator network in distinguishing true and false images and guides the generator network to output more realistic images through the zero-sum game between the generator network and the discriminator network. In this study, the

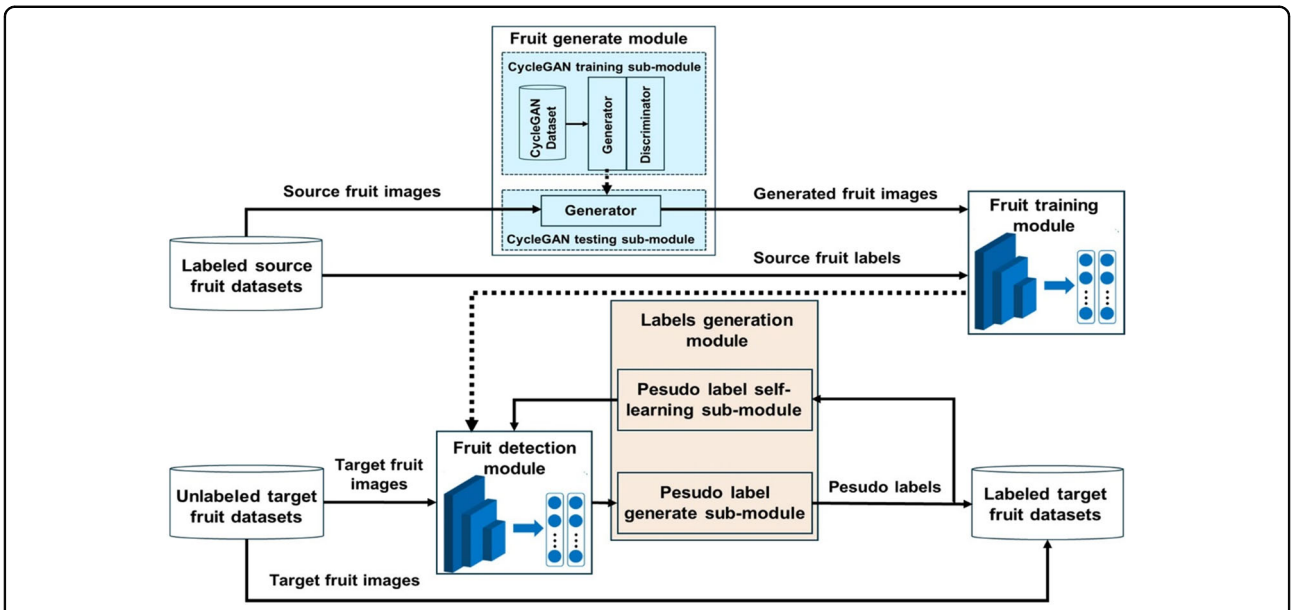

Fig. 1 Workflow of data labeling conversion method. The method mainly includes fruit generate module, fruit training/detection module and label generation module, to relize the automatic data labeling of unlabeled fruit datasets

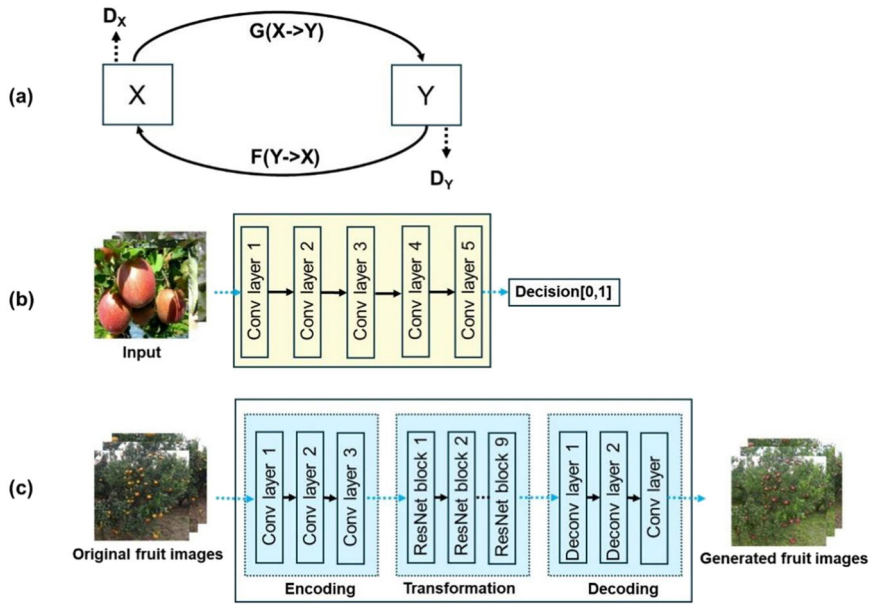

Fig. 2 Image transformation network related components. a represents a mapping function diagram between two image domains X and Y, including two mappings, G:X->Y and F:Y->X, and two discriminators, D_X and D_Y; **b** represents the discriminator network diagram; and **c** represents the generator network diagram

CycleGAN[43] network was deployed to realize image transformation among different species of fruits.

The purpose of the CycleGAN network is to learn the domain mapping between two image domains, X (source domain) and Y (target domain), through unpaired sample images in the dataset, thereby realizing the image transformation between domains without supervision. As shown in Fig. 2a, the CycleGAN network includes two generator networks G and F, for image transformation between two image domains in different directions, and

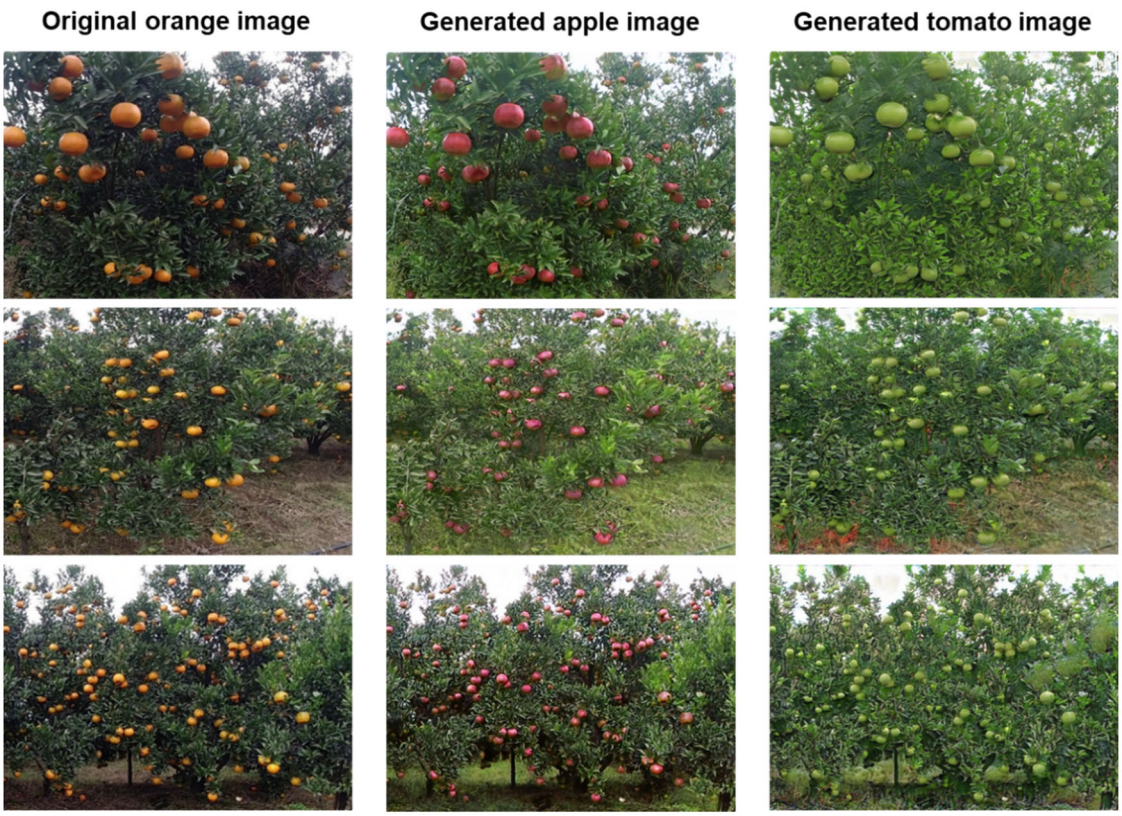

Fig. 3 Examples of source fruit image and generated fake fruit image. The figure shows the image transformation effects at different shooting distances, where the first column shows the source orange image, and the second column shows the generated fake apple image and the third column shows the generated fake tomato image

two discriminator networks D_X and D_Y. The generator network (Fig. 2c) consists of an encoder, a transformer, and a decoder, which operate as follows: first, the source domain image is input into the encoder and the image feature vector is extracted. Afterward, the source domain feature vector is transformed into a target domain feature vector by a transformer, which consists of a residual module constructed of two convolutional layers; this enables the retention of the feature information in the image of the source domain while transforming. Finally, the feature vector of the target domain image output from the transformer is passed through the deconvolution network to reconstruct the low-level features and generate the target domain image. In addition, the discriminator network (Fig. 2b) mainly consists of convolutional layers, which are firstly used to extract image features. The extracted feature vectors are thereupon determined by the one-dimensional output convolutional layer of the last layer and the authenticity of the image is finally determined.

To address the problem of large differences in features between fruits of different species, this paper implements feature transfer between fruits, which is more effective in allowing the model to learn the target fruit features directly. When the CycleGAN network training is completed, the generator network can be used to realize image transformation for different species of fruit images. The operation is as follows. First, train CycleGAN network using different species of source fruit dataset and target fruit dataset, both from CycleGAN dataset, and the image input size of the CycleGAN network is 256*256. Second, using the trained CycleGAN network, according to Eq. (1), transform the source fruit image I_S^i in the dataset D_s into the fake target fruit image I_F^i (as shown in Fig. 3), where w_1 represents the weight parameter of the CycleGAN network. By combining the original labeling information in the dataset D_s, the fake fruit dataset D_F with the source fruit labeling information was constructed.

$$I_F^i = M_1\left(w_1, I_S^i\right), i = 0, 1, 2, ..., N_S \quad (1)$$

Finally, obtain the fruit detection model M_2 by training dataset D_F, which could be applied to the detection task of the dataset D_T^U.

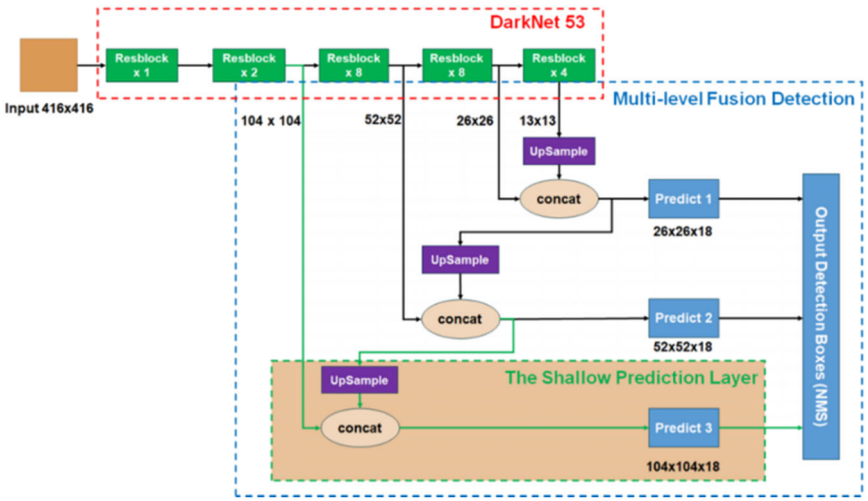

Fig. 4 Structure of the Improved-Yolov3 model. The model mainly consists of Darknet53 backbone, FPN network structure and multi-level detection head

B: Fruit detection network

The detection model applied in this study is grounded on Improved-Yolov3[44]. The model structure is depicted in Fig. 4. Improved-Yolov3 is designed based on the original Yolov3 model, which removes the deep network detection branch with a downsampling rate of 32 and adds a shallow network detection branch with a downsampling rate of 4, fuse the deep and shallow network features by Feature Pyramid Network(FPN) network structure, to improve the small-scale fruit detection performance. More detailed information on Improved-Yolov3 can be found at[44].

C: Pseudo-label generation

The traditional dataset label is based on manual labeling, while pseudo labeling is a machine-generated bounding box similar to manual labeling. This paper proposes a pseudo-labeling approach to generate labels in the dataset D_T^U automatically. Because the fruit features of the fake fruit images generated by the CycleGAN network are more similar to those of naturally grown target fruit images, the model M_2 has some ability to detect real target fruits. Therefore, the labeling information (pseudo label) in the dataset D_T^U can be obtained by the model M_2. The operation is as follows.

First, use the fruit detection model M_2 to obtain the detection bounding box information for real target fruit images in the dataset D_T^U. Thereupon, utilize the acquired detection bounding box as pseudo label of the dataset D_T^U to construct the dataset D_T^L with labeling information automatically and realize the conversion of labeling information between different species of fruit datasets.

D: Pseudo-label self-learning

The detection bounding box obtained by the model M_2 in real target fruit images I_T is used as a pseudo label, and because the model M_2 is trained from the fake fruit dataset D_F, it is prone to the presence of a false detection bounding box in real target fruit images I_T, resulting in noise in the generated pseudo label. Therefore, how to reduce the impact of noise in pseudo labels is one of the main research points in this paper.

In the process of acquiring pseudo labels, the setting of the confidence threshold is related to the quality and quantity of the acquired pseudo labels. When the confidence threshold higher, the acquired pseudo label has a higher probability of correctly labeling the target fruit in the image, while a high confidence threshold leads to a lower number of pseudo labels, and the opposite is also true. Therefore, this paper proposes a pseudo-label self-learning method, which includes a pseudo-label noise filtering operation and a cyclic update operation to reduce the effects of pseudo-label noise, thereby improving the labeling accuracy of pseudo labels, as shown in Algorithm 1. The pseudo-label self-learning method is described as follows.

Pseudo-label noise filtering: First, set the initial confidence threshold θ . The unlabeled target fruit dataset D_T^U is used as the test set input model M_2 to obtain all the detection boxes, as shown in the following equation.

$$\sum_{j=0}^{N_i-1} l_T^{ij} = M_2(w_2, I_T^i, \theta) \tag{2}$$

where l_T^{ij} denotes the jth detection box information of the i^{th} real target fruit image and N_i denotes the total number of detection boxes for the i^{th} real target fruit image, where

$i = 0, 1, 2, \ldots\ldots, N_T - 1$. Subsequently, count the sum of the scores of all detection boxes and calculate the average score S_{aver} according to Eq. (3), filter out the detection boxes below the average score S_{aver}, and the higher score of the detection box is regarded as the pseudo label of the real target fruit dataset D_T^U, as shown in Eq. (4).

$$S_{aver} = \frac{Score(\sum_{i=0}^{N_T-1} \sum_{j=0}^{N_i-1} l_T^{ij})}{\sum_{i=0}^{N_T-1} N_i} \tag{3}$$

$$\sum_{j=0}^{N_i-1} l_T^{ij} = Filter(\sum_{j=0}^{N_i-1} l_T^{ij}, S_{aver}) \tag{4}$$

where the *Score* function indicates that the scores of the acquired detection boxes are summed and the *Filter* function indicates that the detection boxes below the set score value are filtered.

Pseudo-label cycle update: When the model M_2 is fine-tuned using the real target fruit dataset D_T^L for a certain number of epochs, the model M_2 learns the features of the real target fruit image, improves the detection performance of the real target fruit image. At this time, the detection box of the unlabeled real target fruit dataset D_T^U obtained by the model M_2 is more comprehensive and accurate, and the labeling accuracy of the pseudo label is higher. Therefore, the method in this study re-obtains the detection box of the dataset D_T^U by using the current fruit detection model M_2 at certain intervals of training epochs. The pseudo-label information of the unlabeled dataset D_T^U is updated by the aforementioned pseudo-label noise filtering method to improve the labeling accuracy.

Algorithm 1: Pseudo-label Self-learning

Input

Unlabeled Images I_T, labeled dataset D_F, Object Detector M_2, Confidence threshold θ, Number of pseudo-label updates N

Output

Label I_T

1: Initialize M_2 with D_F
2: for $n \leftarrow 1$ to N do:
3: Input M_2 with I_T, obtain I_T
4: Filter noise label I_T based on Eqs. (2)–(4), obtain labeled dataset D_T
5: Update M_2 via fine-tuning with D_T
6: **end**
7: **Output**: label I_T

Experimental setup

This experiment deploys a deep learning framework for model training and testing on a computer platform with an Intel Core i7-8700K CPU processor (32GB of RAM), GeForce GTX 1080Ti GPU graphics card (12GB of video memory), and an operating system with ubuntu18.04LTS, using the Python 3.6.5 programming language to implement the construction, training, and validation of network models under the Pytorch 1.0.0 deep learning framework.

CycleGAN model training: The network was trained using a mini-batch adaptive moment estimation (Adam) optimizer with a momentum factor of 0.5 and a batch size of one. The learning rate for the first 100 training epochs was set to 0.0002, the learning rate for the next 100 training epochs was set to zero with linear recession, and other relevant parameter information from the original paper[43] was applied.

Improved-Yolov3 model training: The detection model is trained in a computer hardware environment with a GPU to improve the convergence rate of model training. Stochastic gradient descent with a mini-batch with a momentum factor was used to train the network. The value of the momentum factor was set to 0.9, the decay was 0.0005, and the batch size was four, the initial learning rate was 0.001, and the learning rate was adjusted using the cosine annealing function. A larger learning rate in the early stage helps the network converge quickly, and a smaller learning rate in the later stage made the network more stable and obtains the optimal solution.

Evaluate metrics

To evaluate the detection performance of the Improved-Yolov3 model, this paper uses Precision, Recall, F1 score, and mAP as the evaluation metrics. A predicted bounding box is considered correct (true positive) if it overlaps more than the intersection-over-union threshold with a labeled bounding box. Otherwise, the predicted bounding box is considered false positive. When the labeled bounding box has an intersection over union with a predicted bounding box lower than the threshold value, it is considered false negative. The standard intersection-over-union threshold value of 0.5 was adopted. The relevant formulae are shown in the following equations.

$$Precision = \frac{Tp}{Tp + Fp} \times 100\% \tag{5}$$

$$Recall = \frac{Tp}{Tp + Fn} \times 100\% \tag{6}$$

$$F1\ Score = \frac{2 \times Precison \times Recall}{Precison + Recall} \times 100\% \tag{7}$$

$$mAP = J(Precison, Recall) \tag{8}$$

where $J(\bullet)$ represents the area calculation function under Precision and Recall curves.

Results

The datasets used in this experiment are described below:

(1)Dataset D_S: contains the images of source oranges and the associated labeling information.

(2)Dataset $D^U_{T\_apple}$: contains the images of real apples without labeling information.

(3)Dataset $D^U_{T\_\text{tomato}}$: contains the images of real tomatoes without labeling information.

Evaluation of datasets D_S and D_F

In this study, the fruit detection model Improved-Yolov3[44] was trained and tested using the dataset D_S and the dataset D_F, respectively. D_S contains source orange dataset D_{S_orange}, and D_F contains fake apple datasets D_{F_apple} and fake tomato datasets D_{F_tomato}. As shown in Table 1, the mAP value obtained by the model Improved-Yolov3 tested in the dataset D_{S_orange} is 95.1%. Because the fake apple image and the fake tomato image were obtained by transforming the orange fruit image in the dataset D_S, the fruit location information is the same in both datasets, with the main divergence being that the underlying features in the image, such as fruit color and texture, are different. After testing, the mAP value of the Improved-Yolov3 model on the dataset D_{F_apple} and D_{F_tomato} are 94.8% and 96.7%, respectively; hence, the difference between the values of each experimental metric on the datasets D_S and D_F is not large, and both have high detection accuracy.

Table 1 Evaluation results for datasets D_S and D_F in the Improved-Yolov3 model

Model	*Datasets*	Precision	Recall	F1 Score	mAP
Improved-Yolov3	D_{S_orange}	0.886	0.923	0.904	0.951
Improved-Yolov3	D_{F_apple}	0.889	0.920	0.904	0.948
Improved-Yolov3	D_{F_tomato}	0.913	0.941	0.927	0.967

Attachment

The following is the attachment related to this paper, mainly including the picture form of the related table.

Adding pseudo labels obtained through different confidence thresholds

As shown in Tables 2, 3, for models obtained from pseudo labels that fine-tune at different confidence thresholds, this experiment was conducted to compare the test results of real apple images and real tomato images. Because there are certain differences in the features between the fake fruit images generated by the CycleGAN network and the natural real-grown fruit images, the model M_2 is fine-tuned using a pseudo-labeling method to reduce the learned feature variability by fitting the feature distribution of the real fruit images. The experiments in this study obtain pseudo labels for the dataset $D^U_{T\_\text{apple}}$ and $D^U_{T\_tomato}$ by setting different confidence thresholds, and the quality and quantity of pseudo labels varied depending on the confidence threshold settings, which impacted fruit detection model M_2. The confidence threshold values ranged from 0.1 to 0.9, and the interval between the values under experimental comparison was 0.1. (The bolded part of the following table indicates the model performance results obtained under the current optimal confidence threshold parameters).

Table 2 Label conversion of orange dataset to apple dataset: the pseudo-labeling method obtaining pseudo labels by setting different confidence thresholds, generating a real apple dataset $D^L_{T\_apple}$ with labeling information, and finally verifying the validity of the generated labels by the model's detection performance

Model	Pseudo label	Conf	Precision	Recall	F1 Score	mAP
Improved-Yolov3	×	None	0.704	0.658	0.68	0.653
	√	0.1	0.724	0.72	0.722	0.769
	√	0.2	0.747	0.746	0.746	0.788
	√	0.3	0.768	0.769	0.768	0.805
	√	0.4	0.783	0.786	0.784	0.828
	√	0.5	0.79	0.8	0.795	0.829
	√	**0.6**	**0.803**	**0.808**	**0.805**	**0.852**
	√	0.7	0.813	0.822	0.817	0.845
	√	0.8	0.815	0.798	0.806	0.843
	√	0.9	0.79	0.796	0.793	0.836

When the real fruit image is tested directly using the model M_2 obtained from the dataset D_F, the mAP value obtained from the real apple and tomato datasets were 65.3% and 71.1%. When using the pseudo-labeling method, as the set confidence threshold increased, the accuracy of the pseudo-label labeling increases, the noise in the pseudo label decreases, and the mAP of the model tends to increase incrementally. When the confidence threshold exceeds a certain value, the mAP value of the model at that time decreases as the confidence threshold value increases, and the reason for the analysis is that the low number of pseudo label with high threshold leads to a decrease in the diversity of features learned, which affects the generalization ability of the model. The model mAP value reached 85.2% when the confidence threshold was 0.6 in the real apple dataset (as shown in Table 2). The model mAP value reached 75.2% when the confidence threshold was 0.5 (as shown in Table 3) in the real tomato dataset, which showed that introducing the pseudo-labeling method improved the fruit detection performance.

Pseudo-label self-learning method to reduce noise labels

There is the effect of noise in the acquired pseudo labels, i.e., incorrect labeling information in the generated pseudo labels affects the training of the fruit detection model. In this paper, pseudo-label noise filtering and cycle update methods are proposed to reduce the impact of noisy pseudo labels. From Tables 4, 5, it is obvious that, as the set confidence threshold increases, the mAP value of

Table 3 Label conversion of orange dataset to tomato dataset: the pseudo-labeling method obtaining pseudo labels by setting different confidence thresholds, generating a real tomato dataset $D^{L}_{T\_tomato}$ with labeling information, and finally verifying the validity of the generated labels by the model detection performance

Model	Pseudo label	Conf	Precision	Recall	F1 Score	mAP
Improved-Yolov3	×	None	0.723	0.725	0.724	0.711
	√	0.1	0.753	0.751	0.752	0.729
	√	0.2	0.754	0.756	0.755	0.732
	√	0.3	0.745	0.747	0.746	0.738
	√	0.4	0.769	0.767	0.768	0.741
	√	**0.5**	**0.77**	**0.769**	**0.769**	**0.752**
	√	0.6	0.765	0.765	0.765	0.748
	√	0.7	0.76	0.759	0.759	0.745
	√	0.8	0.748	0.747	0.748	0.744
	√	0.9	0.705	0.708	0.707	0.688

Table 4 Label conversion of orange dataset to apple dataset: for the pseudo label obtained with different confidence thresholds, the pseudo-label self-learning method is further adopted to reduce the influence of noise in the pseudo label and generate a real apple dataset $D^{L}_{T\_apple}$ with higher quality labels

Model	Pseudo label	Conf	Precision	Recall	F1 Score	mAP
Improved-Yolov3	√	0.1	0.698	0.733	0.715	0.77
	√	0.2	0.747	0.749	0.748	0.79
	√	0.3	0.765	0.771	0.768	0.807
	√	0.4	0.786	0.779	0.782	0.822
	√	0.5	0.793	0.802	0.797	0.828
	√	0.6	0.801	0.796	0.798	0.847
	√	**0.7**	**0.828**	**0.836**	**0.832**	**0.875**
	√	0.8	0.814	0.808	0.811	0.847
	√	0.9	0.793	0.801	0.797	0.838

Table 5 Label conversion of orange dataset to tomato dataset: For the pseudo label obtained with different confidence thresholds, the pseudo-label self-learning method is further adapted to reduce the influence of noise in the pseudo label and generate a real tomato dataset $D^{L}_{T\_tomato}$ with higher quality labels

Model	Pseudo label	Conf	Precision	Recall	F1 Score	mAP
Improved-Yolov3	√	0.1	0.748	0.748	0.748	0.725
	√	0.2	0.757	0.751	0.751	0.731
	√	0.3	0.744	0.749	0.746	0.741
	√	0.4	0.759	0.757	0.758	0.744
	√	0.5	0.766	0.765	0.765	0.764
	√	**0.6**	**0.769**	**0.767**	**0.768**	**0.769**
	√	0.7	0.758	0.757	0.758	0.752
	√	0.8	0.743	0.747	0.745	0.748
	√	0.9	0.731	0.735	0.735	0.717

the fruit detection model M_2 increases and decreases thereupon, mainly due to the effect of the confidence threshold on the quality and quantity of the generated pseudo labels. In the real apple dataset, when the confidence threshold was 0.7, the model mAP value reached 87.5% (as show in Table 4), which is 2.3% higher than the best mAP value in Table 2. In the real tomato dataset, when the confidence threshold was 0.6, the model mAP value reached 76.9% (as show in Table 5), which is 1.7% higher than the best mAP value in Table 5.

Generated datasets labels

From the comparison of the above experimental results, it is clear that the proposed method can generate higher quality label data automatically. In the real apple dataset, the mAP value of the training model reached 87.5% when obtained pseudo-labels with a confidence threshold of 0.7. In the real tomato dataset, the mAP value of the training model reached 76.9% when obtained pseudo-labels with a confidence threshold of 0.6. The above two models have also been applied to visualize apple and tomato detection in real scenarios. As shown in Fig. 5, the image includes target fruit (including apple and tomato) in various scenarios, including complex situations, such as occlusion, shadowing, and underexposure, with the blue box representing the detection results of models. In particular, most of the target fruit in the image can be detected, and the generated detection boxes can well surround the target apples at different locations in the image, which improves the quality of the generated labels, verifies the effectiveness of the proposed method in this study.

Discussion

This paper proposed a new solution to overcome the current problem of high labeling cost for training data acquisition: the automatic labeling of unlabeled fruit datasets. The proposed method could convert labeling between labeled source fruit datasets and unlabeled target fruit datasets to achieve the automatic labeling of target fruit datasets; furthermore, it could be applied for the automatic labeling of other fruit datasets to improve the efficiency of fruit detection work in orchard.

More images of fruit species are currently available in public resources; hence, it is easier to obtain images related to the target fruit species. As shown in Table 6, we collect a large public dataset that included information on access sources, fruit species, and download addresses. It could provide a great deal of data support for subsequent experiments and facilitate experimental testing by other researchers. Therefore, by using the method in this paper, the automatic labeling of other datasets could be completed with solely a small amount of labeling information, thereby saving a great deal of data labeling work and improving fruit inspection efficiency.

In addition, in the practical application of this method, there are certain requirements for the source fruit and target fruit species in the fruit image transformation application: (1) the differences in shape and size between the two fruit species should be as small as possible; and (2) for the source fruit image, the background color features and the fruit color features should be distinguished as clearly as possible. Moreover, in the experimental process, the pseudo labels are mainly obtained by setting the confidence threshold manually, which has the contingency of missing the best confidence threshold. Therefore, more in-depth research on these methods is needed to solve relevant problems, so that the automatic data labeling method could be more effective in a practical level.

Fig. 5 Examples of detection results of apples and tomatoes in real scenarios. The image includes target fruit in different scenarios, where the blue boxes indicate the model detection boxes, and finally, the detection boxes can be used as a ground truth for the unlabeled fruit dataset, enabling the automatic labeling of the dataset

Table 6 Information on some of the current public datasets, including the source of the dataset, the species of fruit, and the associated download URL

Source	Fruit species	Web site
Sa I[32]	Apple, Avocado, Capsicum, Mango, Orange, Rockmelon, Strawberry	http://goo.gl/9LmmOU
Bargoti S[33]	Almonds, Apple, Mango	https://data.acfr.usyd.edu.au/ag/treecrops/2016-multifruit
Koirala,A[35]	Mango	http://hdl.cqu.edu.au/10018/1261224
Kestur,R[36]	Mango	https://github.com/avadesh02
Liang Q[45]	Mango, Almond	https://pan.baidu.com/s/1pdTyVq9PlbhkR2k4Tl5zA
Hani[37]	Apple	http://rsn.cs.umn.edu/index.php/MinneApple
Tsironis V[46]	Tomato	https://github.com/up2metric/tomatOD
Laboroai	Tomato	https://github.com/laboroai/LaboroTomato
Kaggle	Tomato	https://www.kaggle.com/andrewmvd

Conclusion

This paper proposed a domain adaptation method for filling the species gap in deep learning–based fruit detection, which can be applied for the acquisition of labeling information from unlabeled target fruit datasets; this is a new method to solve the high data labeling cost problem. The acceptable accuracy of fruit detection by models trained on the automatically obtained labeled target fruit image showed the effectiveness of the proposed method. With this automatic labeling method, if

there is solely one source fruit dataset with label, the automatic labeling of data from unlabeled target fruit dataset could be realized, saving a large amount of data labeling work. In the future, this method could be applied for the automatic labeling of more fruit datasets to improve the efficiency of orchard work.

It is worth mentioning that there is enormous scope for future research. Notably, we intend to study further on the following aspects: 1) Concerning the image transformation method used in this paper, when the fruit color features and background color features in the source fruit image are similar, the image transformation task is prone to fail. If we successfully solved the transformation problem, the method would be applicable to a wider range of fruit dataset; for this reason, how to solve the image transformation problem captures our interest. 2) During the experiments, pseudo labels are acquired by setting the confidence thresholds manually and are prone to miss the optimal threshold acquisition; hence, we plan to investigate further to obtain the best confidence threshold.

Acknowledgements
This study was partially supported by National Natural Science Foundation of China(NSFC) program U19A2061; Japan Science and Technology Agency(JST) CREST program JPMJCR1512, SICORP Program JPMJSC16H2 and aXis program JPMJAS2018.

Author details
[1]Information Department, Beijing University of Technology, Beijing 100022, China. [2]Institute of Agricultural Resources and Regional Planning, Chinese Academy of Agricultural Sciences, Beijing 100081, China. [3]International Field Phenomics Research Laboratory, Institute for Sustainable Agro-Ecosystem Services, Graduate School of Agricultural and Life Sciences, The University of Tokyo, Tokyo 188-0002, Japan

Author contributions
W.Z., K.C. and W.G. conceived the ideas and designed methodology; K.C., J.W. and Y.S. collected and analyzed the data with the input of W.Z. and W.G.; All authors discussed, wrote the manuscript, and gave final approval for publication.

Data availability
Computer program codes and image data used in this study can be accessed through: https://www.github.com/I3-Laboratory/EasyDAM.

Code availability
Computer program codes and image data used in this study can be accessed through: https://www.github.com/I3-Laboratory/EasyDAM.

Conflict of interest
The authors declare no competing interests.

Received: 14 November 2020 Revised: 16 February 2021 Accepted: 26 March 2021
Published online: 01 June 2021

References

1. L. Jian, Z. Mingrui, and G. Xifeng, A fruit detection algorithm based on r-fcn in natural scene. In Proc. Chinese Control And Decision Conference (CCDC), 487–492, (IEEE, 2020).
2. Ge, Y., Xiong, Y. & From, P. J. Symmetry-based 3d shape completion for fruit localisation for harvesting robots. *Biosyst. Eng.* **197**, 188–202 (2020).
3. Gen´e-Mola, J. et al. Fruit detection, yield prediction and canopy geometric characterization using lidar with forced air flow. *Comput. Electron. Agric.* **168**, 105121 (2020).
4. Wang, Z., Walsh, K. & Koirala, A. Mango fruit load estimation using a video-based mangoyolo—Kalman filter—Hungarian algorithm method. *Sensors* **19**, 2742 (2019).
5. Yu, Y., Zhang, K., Yang, L. & Zhang, D. Fruit detection for strawberry harvesting robot in non-structural environment based on mask-rcnn. *Comput. Electron. Agric.* **163**, 104846 (2019).
6. Dai, N., Xie, H., Yang, X., Zhan, K. and Liu, J. Recognition of cutting region for pomelo picking robot based on machine vision. In Proc. ASABE Annual International Meeting, p. 1, American Society of Agricultural and Biological Engineers, (2019).
7. Liu, W. et al, Ssd: single shot multibox detector. In Proc. European Conference on Computer Vision, 21–37, Springer, 2016.
8. Redmon, J. Divvala, S., Girshick, R. and Farhadi, A. You only look once: Unified, real-time object detection. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, 779–788, (2016).
9. Redmon, J. and Farhadi, A. Yolo9000: better, faster, stronger. In WeProc. IEEE Conference on Computer Vision and Pattern Recognition, 7263–7271, (2017).
10. Farhadi, A., Redmon, J. Yolov3: An incremental improvement[J]. Computer Vision and Pattern Recognition, (2018).
11. Girshick, R. Donahue, J., Darrell, T. and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proc. IEEE Conference on Computer Vision and Pattern Recognition. 580–587, (2014).
12. Wang, X., Shrivastava, A. and Gupta, A. A-fast-rcnn: hard positive generation via adversary for object detection. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2606–2615, (2017).
13. Ren, S. He, K. Girshick, R. and Sun, J. Faster r-cnn: towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems, 91–99, (2015).
14. Balasubramanian, V.N., Guo, W., Chandra, A.L. and Desai, S.V. Computer vision with deep learning for plant phenotyping in agriculture: a survey. Adv. Comput. Commun., 1–26, (2020).
15. Russakovsky, O. et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**, 211–252 (2015).
16. Mu, Y., Chen, T.-S., Ninomiya, S. & Guo, W. Intact detection of highly occluded immature tomatoes on plants using deep learning techniques. *Sensors* **20**, 2984 (2020).
17. Bilen, H., Pedersoli, M. & Tuytelaars, T. Weakly supervised object detection with posterior regularization. *Proc. BMVC* **2014**, 1–12 (2014).
18. Bilen, H., Pedersoli, M. and Tuytelaars, T. Weakly supervised object detection with convex clustering. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, 1081–1089, (2015).
19. Bilen, H. and Vedaldi, A. Weakly supervised deep detection networks. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2846–2854, (2016).
20. Cinbis, R. G., Verbeek, J. & Schmid, C. Weakly supervised object localization with multi-fold multiple instance learning. *IEEE Trans. pattern Anal. Mach. Intell.* **39**, 189–203 (2016).
21. Papadopoulos, D.P. Uijlings, J.R., Keller, F. and Ferrari, V. Training object class detectors with click supervision. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, 6374–6383, (2017).
22. Bellocchio, E., Ciarfuglia, T. A., Costante, G. & Valigi, P. Weakly supervised fruit counting for yield estimation using spatial consistency. *IEEE Robot. Autom. Lett.* **4**, 2348–2355 (2019).
23. Bellocchio, E., Costante, G., Cascianelli, S., Fravolini, M. L. & Valigi, P. Combining domain adaptation and spatial consistency for unseen fruits counting: a quasi-unsupervised approach. *IEEE Robot. Autom. Lett.* **5**, 1079–1086 (2020).
24. Lu, H., Cao, Z., Xiao, Y., Zhuang, B. & Shen, C. Tasselnet: counting maize tassels in the wild via local counts regression network. *Plant Methods* **13**, 79 (2017).
25. Ghosal, S. et al. A weakly-supervised deep learning framework for Sorghum head detection and counting. *Plant Phenomics* **2019**, 1–14 (2019).
26. Chandra, A. L., Desai, S. V., Balasubramanian, V. N., Ninomiya, S. & Guo, W. Active learning with point supervision for cost-effective panicle detection in cereal crops. *Plant Methods* **16**, 1–16 (2020).
27. Settles, B. "Active learning literature survey," tech. rep., University of Wisconsin-Madison Department of Computer Sciences, 2009.

28. Huang, S.-J., Jin, R. and Zhou, Z.-H. Active learning by querying informative and representative examples. In Advances in Neural Information Processing Systems, 892–900, (2010).
29. Wachs, J. P., Stern, H. I., Burks, T. & Alchanatis, V. Low and high-level visual feature-based apple detection from multi-modal images. *Precis. Agric.* **11**, 717–735 (2010).
30. Dubey, S.R., Dixit, P., Singh, N., and Gupta, J.P. Infected fruit part detection using k-means clustering segmentation technique. (2013).
31. Zhang, P. & Xu, L. Unsupervised segmentation of greenhouse plant images based on statistical method. *Sci. Rep.* **8**, 1–13 (2018).
32. Sa, I. et al. Deepfruits: a fruit detection system using deep neural networks. *Sensors* **16**, 1222 (2016).
33. Bargoti, S. and Underwood, J. Deep fruit detection in orchards. In Proc. IEEE International Conference on Robotics and Automation (ICRA), 3626–3633, (IEEE, 2017).
34. Mure¸san, H. & Oltean, M. Fruit recognition from images using deep learning. *Acta Univ. Sapientiae, Inform.* **10**, 26–42 (2018).
35. Koirala, A., Walsh, K., Wang, Z. & McCarthy, C. Deep learning for real-time fruit detection and orchard fruit load estimation: bBenchmarking of 'mangoyolo'. *Precis. Agric.* **20**, 1107–1135 (2019).
36. Kestur, R., Meduri, A. & Narasipura, O. Mangonet: a deep semantic segmentation architecture for a method to detect and count mangoes in an open orchard. *Eng. Appl. Artif. Intell.* **77**, 59–69 (2019).
37. H¨ani, N., Roy, P. & Isler, V. Minneapple: a benchmark dataset for apple detection and segmentation. *IEEE Robot. Autom. Lett.* **5**, 852–858 (2020).
38. Goodfellow, I. et al, Generative adversarial nets. In Advances in Neural Information Processing Systems, 2672–2680, (2014).
39. Stein, G. J., & Roy, N. Genesis-rt: generating synthetic images for training secondary real-world tasks. In Proc. IEEE International Conference on Robotics and Automation (ICRA). 7151–7158, (2018).
40. Zhang, J. et al. Vr-goggles for robots: Real-to-sim domain adaptation for visual control. *IEEE Robot. Autom. Lett.* **4**, 1148–1155 (2019).
41. Roy, P., Häni, N., & Isler, V. Semantics-aware image to image translation and domain transfer. arXiv preprint arXiv:1904.02203. (2019).
42. Valerio Giuffrida, M., Dobrescu, A., Doerner, P., & Tsaftaris, S. A. (2019). Leaf counting without annotations using adversarial unsupervised domain adaptation. In Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops (2019).
43. Zhu, J.Y., Park, T., Isola, P. and Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proc. IEEE International Conference on Computer Vision, 2223–2232, (2017).
44. Wang, J. et al, A deep learning-based in-field fruit counting method using video sequences. https://www.plant-phenotyping.org/CVPPP2020-Programme.
45. Liang, Q. et al, A real-time detection framework for ontree mango based on ssd network. In Proc. International Conference on Intelligent Robotics and Applications, 423–436, (Springer, 2018).
46. Tsironis, V., Bourou, S. & Stentoumis, C. Tomatod: evaluation of object detection algorithms on a new real-world tomato dataset. *Int. Arch. Photogramm., Remote Sens. Spat. Inform. Sci.* **43**, 1077–1084 (2020).