Computers and Electronics in Agriculture 197 (2022) 107000

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

# A novel apple fruit detection and counting methodology based on deep learning and trunk tracking in modern orchard

Fangfang Gao [a], Wentai Fang [a], Xiaoming Sun [a], Zhenchao Wu [a], Guanao Zhao [a], Guo Li [a], Rui Li [d], Longsheng Fu [a,b,c,e,*], Qin Zhang [e]

[a] College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling 712100, China
[b] Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture and Rural Affairs, Yangling, Shaanxi 712100, China
[c] Shaanxi Key Laboratory of Agricultural Information Perception and Intelligent Service, Yangling, Shaanxi 712100, China
[d] Suide County Lanhuahua Ecological Food Co., Ltd., Suide, Shaanxi 718000, China
[e] Center for Precision and Automated Agricultural Systems, Washington State University, Prosser, WA 99350, USA

ARTICLE INFO

*Keywords:*
Yield estimation
Orchard video
Apple detection
Object tracking
Fruit position
Fruit ID

ABSTRACT

Accurate count of fruits is important for producers to make adequate decisions in production management. Although some algorithms based on machine vision have been developed to count fruits which were all implemented by tracking fruits themselves, those algorithms often make mismatches or even lose targets during the tracking process due to the large number of highly similar fruits in appearance. This study aims to develop an automated video processing method for improving the counting accuracy of apple fruits in orchard environment with modern vertical fruiting-wall architecture. As the trunk is normally larger than fruits and appears clearly in the video, the trunk is thus selected as a single-object tracking target to reach a higher accuracy and higher speed tracking than the commonly used method of fruit-based multi-object tracking. This method was trained using a YOLOv4-tiny network integrated with a CSR-DCF (channel spatial reliability-discriminative correlation filter) algorithm. Reference displacement between consecutive frames was calculated according to the frame motion trajectory for predicting possible fruit locations in terms of previously detected positions. The minimum Euclidean distance of detected fruit position and the predicted fruit position was calculated to match the same fruits between consecutive video frames. Finally, a unique ID was assigned to each fruit for counting. Results showed that mean average precision of 99.35% for fruit and trunk detection was achieved in this study, which could provide a good basis for fruit accurate counting. A counting accuracy of 91.49% and a correlation coefficient $R^2$ of 0.9875 with counting performed by manual counting were reached in orchard videos. Besides, proposed counting method can be implemented on CPU at 2 ~ 5 frames per second (fps). These promising results demonstrate the potential of this method to provide yield data for apple fruits or even other types of fruits.

## 1. Introduction

Obtaining an accurate counting of fruit in a certain part on a tree could provide critical information for yield estimation and support effective precision orchard management. Yield estimation in an apple orchard is important for growers as it facilitates efficient utilization of resources and improves benefits (Bargoti and Underwood, 2017; Fountas et al., 2020; Koirala et al., 2019; Wu et al., 2022). Despite the importance, there is a lack of reliable and accurate automated fruit counting methods, and growers today rely on manual counting to support their decision making for field operations (Payne et al., 2014; Dorj et al., 2017; Apolo-Apolo et al., 2020; He et al., 2022). As manual counting is labor-intensive and high-cost, developing an automatic fruit counting method with high accuracy and reliability has been highly desired by fruit growers.

In recent years, researchers have tried to develop various methods for accurately detecting fruits. Some of widely studied methods include K-means clustering (Jiao et al., 2020), simple linear iterative clustering (Liu et al., 2019b), circular Hough transform (Nguyen et al., 2016), directional gradient histogram (Li et al., 2021), and many more (Grilli et al., 2021). Such methods rely on the color, shape, and texture characteristics of objects to detect the objects, which were found being

* Corresponding author at: College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling 712100, China.
*E-mail addresses:* fulsh@nwafu.edu.cn, longsheng.fu@wsu.edu (L. Fu).

https://doi.org/10.1016/j.compag.2022.107000
Received 7 December 2021; Received in revised form 18 April 2022; Accepted 20 April 2022
Available online 26 April 2022

challenging to achieve a robust result in natural orchard environment (Qureshi et al., 2017; Wang et al., 2018; Zhang et al., 2020). Recent studies revealed that deep learning methods based on convolutional neural networks with advanced semantic expression ability could improve the robustness because they could autonomously learn shallow and deep features of objects (Lin et al., 2020; Zhou et al., 2020; Song et al., 2021; Wu et al., 2021; Wang et al., 2022). Compared with the two-stage detection network with higher detection accuracy, the one-stage detection network has faster detection speed (Gené-Mola et al., 2019; Gao et al., 2020; Li et al., 2022). And in the one-stage detection networks, YOLO (You Only Look Once) can obtain higher detection accuracy in fruit detection (Kuznetsova et al., 2020; Fu et al., 2021). Lu et al. (2022) applied YOLOv4 to detect apples and obtained a detection rate of 92.6%, which was 8.3% and 7.4% higher than using Faster RCNN and SSD, respectively. Gao et al. (2021) detected apples by YOLOv4-tiny, and reached 94.47% detection rate with 0.018 s to process an image with 1280 × 720 resolution. Therefore, YOLOv4-tiny (Wang et al., 2020) is considered to have the potential to achieve high-precision detection at a faster detection speed (Wang et al., 2018).

Some studies have been achieved to count the number of fruits on a tree based on image detection using deep learning. Stein et al. (2016) employed Faster R-CNN based on VGG16 to detect fruits from multiple angles images of a mango tree, and applied trajectory data provided by a navigation system to establish correspondences between images for fruit counting. Compared with the results based on manual counting, the error rate of a single mango tree was only 1.36%. Koirala et al. (2019) proposed a network "MangoYOLO" to detect mango fruits on all images collected in an orchard, and estimated orchard fruit load by counting the number of detected fruits. But the resulting estimate was 4.6% to 15.2% higher than the actual load. Apolo-Apolo et al. (2020) trained a long short-term memory model based on images obtained by different sides of citrus tree for estimating the yield of citrus, and resulted in an approximate error of 4.53% between actual and estimated yields per tree. Although these methods obtained good fruit counting results based on orchard images, they all need to manually control shooting areas to avoid overlapping areas in the orchard series images. Video of multiple fruit trees, which can be easily obtained by a ground vehicle, is a convenient alternative that can help to quickly count fruits. However, to count fruits in the video, methodologies are needed to correlate the same fruits between consecutive video frames to avoid repeat counting.

Tracking, a fundamental task in any video application requiring reasoning about objects of interest, can establish object correspondences between video frames. Given the position of any object of interest in the first frame of video, the purpose of visual object tracking is to estimate its position in all subsequent frames with the highest possible accuracy (Wang et al., 2019a). The tracking approach is widely applied in the fields of military aerospace, security surveillance, and intelligent driving (Lee et al., 2019; Ngo et al., 2019; Wawrzyniak et al., 2019). In recent years, tracking algorithms based on discriminative correlation filter (DCF) have been widely applied due to relatively stable tracking performance and fast speed (Kalal et al., 2012; Danelljan et al., 2014). A CSR-DCF (Channel Spatial Reliability-Discriminative Correlation Filter) tracking algorithm has been proposed on a recent tracking benchmark, which achieves good performance by introducing the channel and spatial reliability concepts to DCF tracking (Lukežič et al., 2018). Farhodov et al. (2019) integrated Faster R-CNN with CSR-DCF algorithm to track pedestrian and vehicle and demonstrated outstanding real-time detection and tracking performance.

Reported researches on video-based fruit counting were mostly implemented by tracking all fruits detected in video frames. Liu et al. (2019a) developed a mango fruit counting system based on a Kalman filter to track all detected fruits, and reached an $R^2$ value of 0.88 related to the actual number. However, this research needs to reconstruct landmarks with semantic structure from motion (SfM) features to identify repeated tracked fruits. Wang et al. (2019b) also employed the Kalman filter to track all mango fruits detected using the 'MangoYOLO' network and reported a bias corrected Root Mean Square Error (RMSE) of 18.0 fruit/tree. Vasconez et al. (2020) applied Gaussian estimation multi-object tracking algorithm developed by Milan et al. (2017) to track all apple fruits detected by Faster R-CNN with InceptionV2, which reached a 93% counting performance with a detection rate of 54%. The above-mentioned researches need to track all detected fruits, which requires a lot of computing resources and even lose the accuracy of multi-target tracking. Therefore, counting fruits in the video through resource-saving single-object tracking is more promising for actual field applications.

By tracking the trunk of an apple tree trained in vertical fruiting-wall architecture, the motion trajectories of all relatively stationary objects in video frames, including trunks, branches, leaves, and fruits, can be obtained. Existing video tracking technology mainly focuses on tracking some large objects with obvious characteristics, such as pedestrians (Wojke et al., 2018), cars (Li et al., 2014), or humans with actions (Jain et al., 2014). However, apple fruits are small objects with similar characteristics to each other, which makes it difficult to track them in an unrestricted environment. Because trunk with obvious characteristics is also an essential part of a fruit tree just like fruits, and larger than the apple fruit. The trunk is thus more suitable for tracking than the fruit in the orchard video. In modern vertical fruiting-wall architecture, trees have obvious characteristics and are planted close together with inter-plant spacing generally ranging from 0.3 to 1.5 m, while inter-row spacing varies typically from 2.5 to 4.0 m (Fu et al., 2020a; Gao et al., 2020). This makes it possible to detect the trunk in each frame of the acquired video. Therefore, the relationship between the fruits of consecutive frames of the video can be established by tracking the trunk, thereby eliminating the repeated counting of fruits.

This study aimed to create an apple video automatic counting method based on YOLOv4-tiny detection network and single-object tracking algorithm for accurately accounting fruits and estimating the yield in apple orchards. This work is organized as follows: Section 2 will describe the design of this proposed method, Section 3 will present the obtained results of fruit detection, object tracking, and fruit counting over frame sequences, and discuss the relevant issues, and Section4 will present the conclusions acquired from this study.

## 2. Materials and methods

As stated, this study has focused on developing an image processing system for accurately counting apple fruits in vertical fruiting-wall tree architecture. This system was constructed by a portable device used for acquiring images and videos from apple orchards, and an image processing pipeline being specifically proposed for detecting, tracking, and counting fruits in this particular application. The pipeline is shown in Fig. 1. The core of this image processing pipeline was a YOLOv4-tiny based object detection network and a CSR-DCF-based tracker. To count fruits on one side of a specific row of trees, the pipeline took in videos recorded from one side of tree row. Firstly, fruits and trunks were detected using YOLOv4-tiny in frame *i*. The detection result of the trunk was set as the input of the CSR-DCF-based tracker to obtain the motion trajectory for all objects that are relatively static to the trunk, including fruits. Then reference displacement between video frames could be calculated to predict the fruit position. The matching of the same fruit between consecutive video frames was achieved by finding the detected fruit and the predicted fruit with the minimum Euclidean distance. Based on detected fruit positions, fruit IDs (Identity Documents) were given to complete fruit counting. Finally, the ID with a maximum value of the fruit in the last frame of the video was obtained as the number of fruits in the video.

### 2.1. Data acquisition

Image and video data were collected using the developed portable device in a commercial fruiting-wall 'Scifresh' apple orchard with a

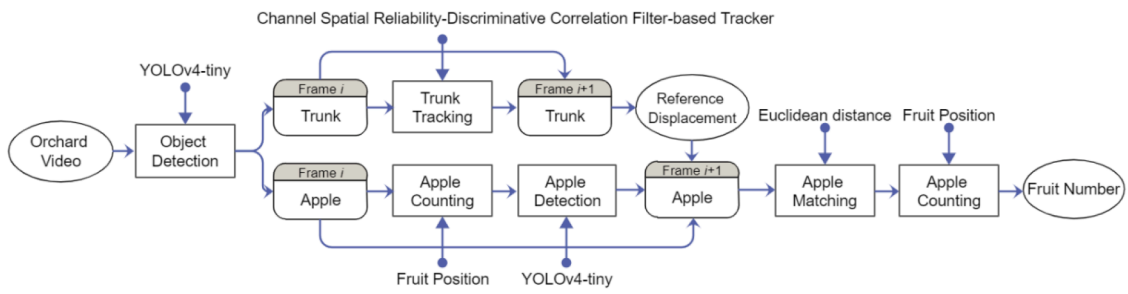



**Fig. 1.** Apple fruit counting pipeline developed by integrated detection network and tracking algorithm.

dense-foliage canopy. A Microsoft Kinect V2 sensor (Microsoft Inc., Redmond, WA, USA), a camera clamping device, and a support frame were mounted on a field remote control vehicle to form the device for acquiring fruit-on-plant images and videos. The resolution and field of view of the color image acquired by Kinect V2 are 1920 × 1080 and 84.1 × 53.8, respectively (Fu et al., 2020b). All image and video data were acquired in a commercial apple orchard with vertically trellis-trained tree architecture (Fig. 2) near Prosser, WA, USA. The height of the trees was about 4.0 m with an inter-tree spacing of 1.5 m and inter-row spacing of 2.7 m.

A total of 800 RGB (Red, Green, and Blue) static images and 20 living videos of tree canopy were acquired in two consequent harvesting seasons of 2017 and 2018. Apple images were obtained using the Kinect V2 sensor with a JPEG image format for developing an algorithm for fruit and trunk detection. The resolution of the acquired images and videos is 1920 × 1080 pixels. During the canopy video acquisition, the vehicle was driven along the rows of apple trees and was tried to maintain at an almost constant speed. The time of video acquisition was random, which make differences in the numbers of trees and fruits contained in the video. The total numbers of fruits and trunks in each video were first counted manually by three different operators. Then the average value was calculated as the ground truth. Every fruit in the video was counted except for the fruits in the background. Uncertainty in manual count typically occurred around dimly lit fruit on the canopy margin and background, and around highly occluded fruit (Qureshi et al., 2017). The relative error for manual counts of the video set by three different operators was 2.7 fruits per video, which indicated the reliability of the ground truth.

### 2.2. Data building

Dataset was constructed using image data and video data to train and test network performance. The overall image data were randomly divided into a raw training dataset (80% of the images, 640 images) and a testing dataset (20% of the images, 160 images) to train and test the YOLOv4-tiny model, respectively. For each image, an XML annotation file was created containing image attributes (name, width, height) and object attributes (class name, object bounding box co-ordinates). Fruits and trunks were labeled using rectangular boxes with different class tags in the XML file. Because targets in the video are far from their full shapes when they are at the boundary, it will make matching and counting difficult. All targets that are at the boundary of the image were not labeled. While all the videos were used to test the performance of the counting algorithm.

Data augmentation was applied to enlarge the image training dataset which helped to reduce over-fitting in supervised learning algorithms. To achieve sensitive detection of fruits and trunks in the orchard, this study considered most kinds of interference that may occur when detecting the objects. Data augmentation, including brightness transformation (Fig. 3a), adaptive histogram equalization (Fig. 3b), motion blur transformation (Fig. 3c), and image mirroring in horizontal axis (Fig. 3d) on the dataset, were implemented. The original 800 images were augmented to a total of 6400 images. Although many studies have proved that rotation can improve network performance, considering that the angle of the trees in the image generated by rotation does not conform to the actual situation, this study did not choose rotation as the augmentation method.

### 2.3. YOLOv4-tiny network for fruits and trunks detection

In terms of fruits and trunks detection using CNN-based network, YOLOv4-tiny can be used to develop a network model with excellent detection accuracy and speed. Fig. 4 shows the structure of YOLOv4-tiny created for this image-based fruit counting application. YOLOv4-tiny uses 36 layers (starting from 0 and removing input and output layers) instead of more than a hundred required for YOLOv4. Thus, it could achieve a quicker training process attributing to less computational load without too much deterioration in detection performance (Montalbo, 2020). The 1920 × 1080 images are automatically compressed and stretched to 416 × 416 images by the network without preserving the aspect ratio. The extractor of this developed structure took the generated 416 × 416 image at the input, with its backbone came with

**(a)** **(b)**

**Fig. 2.** Experimental orchard of this work. **(a)** Apple trees of the 'Scifresh' variety during harvest season; **(b)** Image obtained in the orchard.

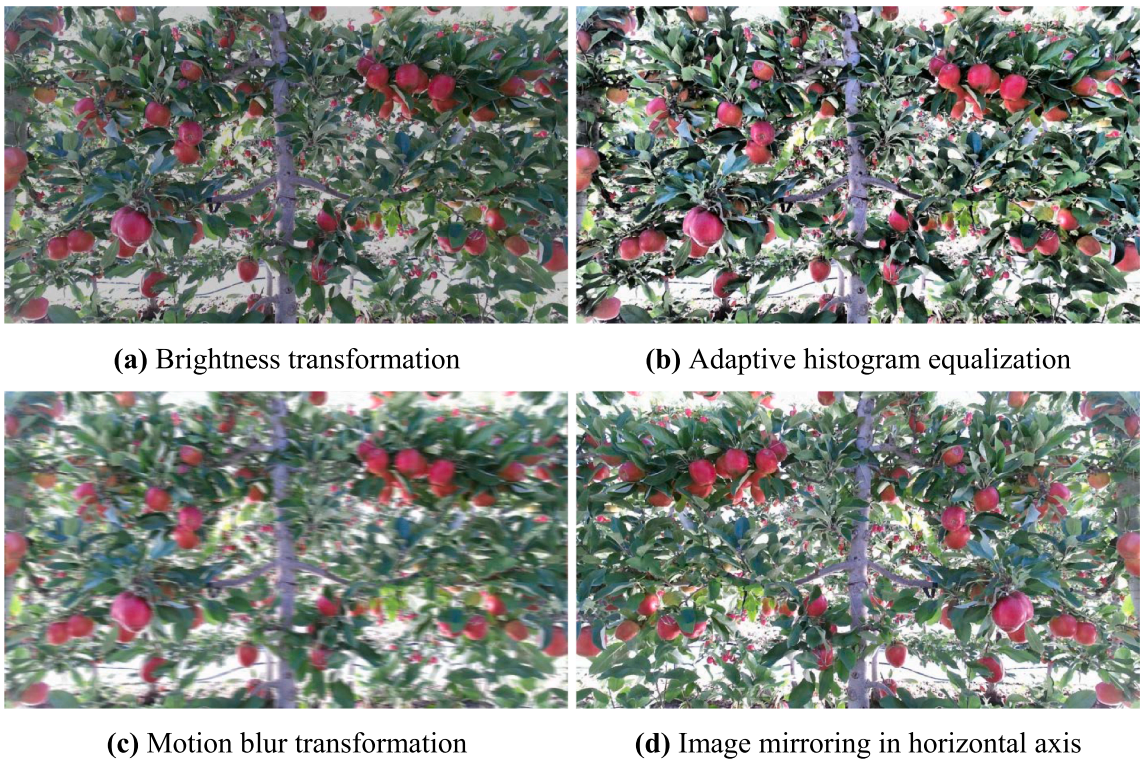



**Fig. 3.** Example of augmentations.

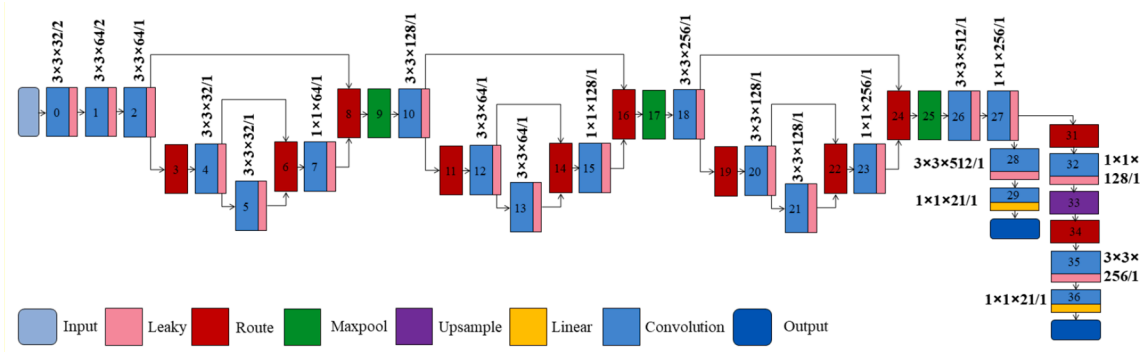



**Fig. 4.** Structure of YOLOv4-tiny network. The number of each layer in the network represents the number of layers of the network. The output feature map size of each layer in the network is expressed as "Width × Height × Number of filters/stride".

interchanging 3 × 3 and 1 × 1 receptive filters to stride over the input for producing a new set of feature maps or filters that pass through the network. Detector used both the feature map of the final level and the feature map of the previous level to detect various sized objects by applying the concept of feature pyramid network. Because many convolutional layers cannot be applied to lower-level feature maps, it is difficult to extract enough feature maps. Therefore, YOLOv4-tiny employs only the two feature maps for object detection, unlike the YOLOv4.

### 2.4. CSR-DCF-based tracker

The CSR-DCF-based tracker is a key component of the proposed counting method. Based on channel spatial reliability, it offers a promise on achieving excellent performance on object tracking in real-time on a CPU (Lukežič et al., 2018). To build the developed fruit counter based on CSR-DCF algorithm, the YOLOv4-tiny network was combined to provide the required object information. The detection network provided the tracker with the position of the object to be tracked. The tracker trained the DCF template by two simple standard features of the object (HoGs and Colornames) to predict where the object may appear in the next frame. The new position of the object where the maximum response point is located was applied to train and update the template for subsequent prediction. The predicted position with a large deviation in the video would be adjusted based on the detection result of the current video frame. With the great afford of deep learning-based object detection can avoid misclassification and loss of objects to a certain extent, thereby ensuring the accuracy of the tracking.

### 2.5. Fruit counting algorithm

Another major function this developed counter needed to perform was to match the same fruits between consecutive video frames. This task was completed by tracking the trunk in those consecutive frames. In the matching process, the pixel position of all detected objects (including fruits and trunks) with a confidence higher than 0.5 would be first extracted. The fruits and trunks were distinguished based on the

class tag index, of which the first trunk in video motion direction was selected as the tracking object. The tracker was initialized using the detected position of the selected tracking object to get the predicted position for the next frame. Then the reference displacement between consecutive video frames can be calculated by the predicted position and the initial position. Because all fruits and trunks in the same frame have the same trajectory, they have the same displacement. Therefore, the predicted position obtained by adding the reference displacement to the initial position of fruit should ideally be the same as the position of a certain fruit detected in the next frame, so that the same fruit in consecutive frames of the video can be successfully matched.

The existence of detection and tracking deviation made the predicted position and the detection position not exactly the same. Although both the YOLOv4-tiny network and CSR-DCF-based tracker had good performance, the position of the fruit and trunk output by them cannot completely coincide with the ground truth. Then, the predicted fruit position and the detected fruit position will have a certain distance. However, there should be a minimum distance between the predict position of the fruit and the detected position of the same fruit. Therefore, the Euclidean distance in a two-dimensional space can represent the similarity of the fruits in consecutive video frames. The two fruits with the minimum Euclidean distance have the greatest similarity and will be regarded as the same fruit.

The reference displacement is the key value to calculate the Euclidean distance. Inaccurate values of the reference displacement may cause errors in the calculation of the minimum Euclidean distance. Excessive tracking deviation is the root cause of inaccurate reference displacement calculation. The overlap between the predicted area and detected area reflects the magnitude of the tracking deviation. The overlap rate will decrease as the tracking deviation increases. When the degree of overlap drops to a specified threshold, the position of the tracked trunk will be promptly adjusted according to the detected trunk position. At the same time, to avoid the inability to obtain the reference displacement due to the tracking failure, the trunk that reaches the boundary of the video frame will no longer be the tracking object, while the other trunk detected in the current frame will be regarded as a new tracking object.

The detection network and the tracking algorithm were integrated for developing the counting method. The fruit detection and trunk tracking results of each frame were set as the input to the developed fruit counter. The position of a tracked object was expressed using pixel coordinates (*x, y, w, h*), where *x* and *y* respectively refer to the horizontal and vertical coordinates of the upper left corner of the target detection rectangle, *w* is the width of the rectangle, and *h* is the height of the rectangular box. Let $B_a^i$ be the bounding box of fruit for frame *i*, $B_t^i$ be the bounding box of trunk for frame *i*. Let (x', y') be the central coordinates of the bounding box, which can be calculated in Eq. (1) below:

$$(x', y') = \left(x + \frac{w}{2}, y + \frac{h}{2}\right) \tag{1}$$

The steps of fruit counting from a video clip are as follows (assume started at frame 1):

For frame 1, the processing steps include ($S_{ij}$ refers to processing steps, *i* refers to video frame, *j* refers to specific step):

$S_{11}$: Detect video frame by YOLOv4-tiny to obtain $B_a^1$ and $B_t^1$;

$S_{12}$: Assign fruit IDs incrementally starting from 1 according to the video motion direction, put current count results into a list *L*, and obtain current fruit number *N*;

$S_{13}$: Extract the bounding box $B_{tu}$ of trunk *u* with the smallest abscissa to initialize the tracker.

For frame 2, the processing steps include:

$S_{21}$: Detect video frame by YOLOv4-tiny to obtain $B_a^2$ and $B_t^2$;

$S_{22}$: Assign fruit IDs incrementally starting from $N + 1$ according to the video motion direction, put the current count results into the list *L*, and obtain the current fruit number *N*;

$S_{23}$: Extract the new bounding box coordinates $B_{tu}$' of the trunk *u* output by the tracker in current frame;

$S_{24}$: Calculate the overlap rate (OV) between $B_{tu}$' and all trunk detection bounding boxes $B_t^2$ based on Eq. (2), if the maximum overlap is less than 40% (Overlap threshold of [20%, 30%, 40%, 50%, and 60%] were experimented, which of 40% to be the most appropriate threshold for detecting overlaps because it can obtain the highest counting accuracy.), replace $B_{tu}$' with the $B_t^2$ with the largest overlap rate to update the tracker;

$S_{25}$: Calculate the coordinate difference between the $B_{tu}$' of the trunk *u* in the current frame and the $B_{tu}$ of the trunk *u* in the previous frame as the reference displacement (RD).

$S_{26}$: Calculate the Euclidean distance (Ed) between fruits detected in consecutive video frames based on Eq. (3), where the two fruits with the minimum Euclidean distance are considered the same fruit;

$S_{27}$: Change the ID of the same fruit in the current frame to the ID of the previous frame, update the list *L* and the number of current fruits *N*.

$$\mathrm{OV}_{B_t^i}^{B_{tu}'} = \frac{area\left(B_{tu}' \cap B_t^i\right)}{area\left(B_{tu}' \cup B_t^i\right)} \tag{2}$$

$$\mathrm{Ed} = \sqrt{\left(x_{B_a^{i-1}+RD} - x_{B_a^i}\right)^2 + \left(y_{B_a^{i-1}} - y_{B_a^i}\right)^2} \tag{3}$$

Repeat the steps of the frame 2 till the current tracking trunk *u* reaches the boundary of the video frame, then change another trunk *v* as the new tracking target. Repeat the above steps until the end of the video, and output the current number of fruits *N*.

### 2.6. Experimental platform and related settings

Platform for training the detection network included a desktop computer equipped with a CPU of Intel Core i5-6400 (2.70 GHz), 16 GB of RAM, and an NVIDIA GTX 1080 8 GB GPU, running on a Windows 10 64-bit system. Software tools included CUDA 9.0, cuDNN 7.1.3, Microsoft Visual Studio 2015, CMake-3.16, Python 3.6, and OpenCV 3.1.0. Experiments were implemented on Darknet framework. The network input size was 416 × 416, with a batch size of 64. Stochastic gradient descent was applied for training with a momentum of 0.9 and a weight decay of 0.0005. Initial values of 0.001 and 50,000 were set as the learning rate and iterations of the network, respectively. Transfer learning technique was performed to train the network by importing weights from previous training on the COCO dataset, which contains 91 object types with 2.5 million labeled instances in 328k images (Lin et al., 2015).

Platform for debugging the counting algorithm included a laptop computer equipped with a CPU of Intel Core i7-8565U (1.80 GHz), 8 GB of RAM, and an NVIDIA GeForce MX250 2 GB GPU, running on a Windows 10 64-bit system. Software tools included CUDA 10.0, cuDNN 7.6.5, Microsoft Visual Studio 2017, Python 3.8, and OpenCV 4.4.0. The proposed method was developed based on Python language with OpenCV library. The CSR-DCF algorithm and Darknet framework were integrated with the OpenCV library of 4.4.0 version, which supports this research to establish the CSR-DCF-based tracker and load the YOLOv4-tiny network. The network layers of 30 and 37 were set as the YOLOv4-tiny network output layer, forward propagation to obtain the detection results of fruits and trunks. The first trunk detected in the direction of video movement was selected as the tracking object to create a tracker for follow-up work.

### 2.7. Evaluation indicators

The performance of the fruit and trunk detection network trained in this study was assessed using mean average precision (mAP). All samples were divided into four types according to the combinations of the true and predicted class: TP (true positive), FN (false negative), FP (false positive), and TN (true negative). The P (Precision) and R (Recall) of the detection network were defined in Eq. (4) and Eq. (5) according to the classification of the samples. AP was an indicator that reflects the global performance of the network, which was calculated in Eq. (6) by P (Precision) and R (Recall). The mAP is calculated from AP values of fruit and trunk detection (Suo et al., 2021), which is defined by Eq. (7).

$$P = \frac{TP}{TP + FP} \tag{4}$$

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

$$AP = \int_0^1 P(R)dR \tag{6}$$

$$mAP = \frac{1}{2}\left(AP_{apple} + AP_{trunk}\right) \tag{7}$$

Another two metrics MIDE (Mean ID calculation Error) and RMSE were proposed to evaluate the performance of fruit counting in video frames, as shown in Eq. (8) and Eq. (9). Besides, $P_c$ (counting accuracy) was defined in Eq. (10) to evaluate the performance of fruit counting in the video.

$$MIDE = \frac{1}{n}\sum_{i=1}^{n}\frac{N_{si}}{N_{ci}} \tag{8}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(N_{ci} - N_{gi}\right)^2} \tag{9}$$

$$P_c = \left(1 - \frac{|P_t - N_t|}{N_t}\right) \times 100\% \tag{10}$$

where $N_{si}$, $N_{ci}$, and $N_{gi}$ are the number of fruits whose ID has switched in the frame $i$ compared to the previous frame, the number of counted fruits in the frame $i$ using the proposed method, and ground truth of frame $i$, respectively. $P_t$ and $N_t$ refer to the number of counted fruits and ground truth of video, respectively. The ground truth is the average of manual counting of the video set by three different operators. The $n$ in equations refers to the number of video frames.

## 3. Results and discussion

In this section, the relationship between information obtained by the proposed method and real information measured on each video was evaluated and verified. Initially, P, R, AP, and mAP were calculated to evaluate the detection model of fruit and trunk trained based on the YOLOv4-tiny network. Afterward, the developed counting method was applied to orchard videos for fruit counting. The correctness of fruit IDs obtained based on the proposed counting method in each frame of the video is evaluated according to the fruit ID manually labeled. Then, a multi-object tracking method based on Deep Sort was developed to count videos for comparing with the proposed method. Finally, the shortcomings of this research and the future direction of the research are analyzed.

### 3.1. Performance of the detection network

The developed detection model was evaluated using the test dataset with ground truth. Fig. 5 reveals that the training loss curve of the YOLOv4-tiny based model was converged within 50,000 iterations. Loss value is generally stable when the number of iterations reaches 43,000 iterations and approaches the lowest value of 0.90. The curve demonstrates that the network efficiently learns the features with good convergence ability. The size of the trained weight is only 22.4 MB, which is beneficial for offline field real-time application and further application in portable devices. The confidence threshold was set to 0.25 when testing the images. Out of a total of 37,642 targets of interest in these 1,280 testing images, the trained model successfully detected 37,376 objects (TP), with false positive of 4,036 and false negative of 266 (Table 1). Further analysis found that the TP for trunk detection was 1656 out of 1675 showed objects in those 1,280 images (AP rate of 99.10%), and fruit detection TP reached 35,720 out of a total 35,967 objects in testing images (AP rate of 99.59%). Such high AP rates made the mAP reached a high value of 99.35% which provides an excellent basis for accurate fruit counting. Besides, YOLOv4-tiny took only 0.022 s on average to detect fruits or trunks in one image of 1920 × 1080 pixel resolution, indicating that the detection model developed here could quickly and accurately detect fruits and trunks, and achieved the aimed goal of this study.

The YOLOv4-tiny model achieves good performance for detecting both fruits and trunks in natural orchards. 99.31% (35720 out of 35967) of the fruits are successfully detected from the background, although there are FN and FP detections. FP detection is caused by that the object on the frame boundary is not labeled in the ground truth but is successfully detected, which is beneficial to the successful counting of fruits. Moreover, the FN detection boxes will not make a significant

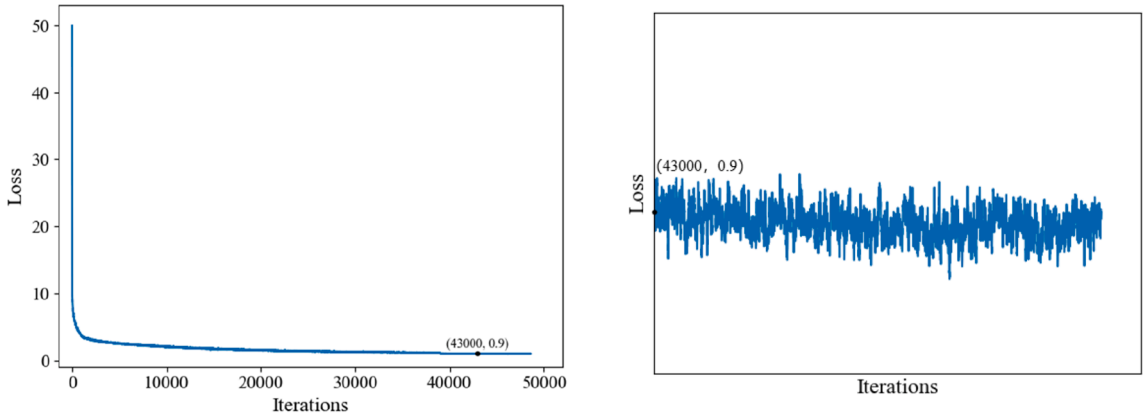



**Fig. 5.** Training loss curve of YOLOv4-tiny network. **(a)** Full training loss curve during training; **(b)** Local training loss curve during training.

**Table 1**
The detection results on the testing dataset.

| Object | TP | FP | FN | TN | P/% | R/% | AP/% | mAP/% | Speed/ (ms/image) |
|---|---|---|---|---|---|---|---|---|---|
| Fruit | 35,720 | 3972 | 266 | 15,663 | 90.26 | 99.29 | 99.59 | 99.35 | 22 |
| Trunk | 1656 | 63 | | | | | 99.10 | | |

negative effect on further fruit counting, because the fruit counting method has great potential to correctly count the fruit by detecting multiple successive frames that all containing the incorrectly classified fruits (Fig. 6). Because the trunks are large objects in the orchard image and have obvious features, 98.87% (1656 out of 1675) of the trunks in the testing dataset are detected, even some of them are incomplete at the boundary of the image. Although the detection rectangle of the trunk is not always the minimum bounding rectangle of the target, it will not affect the performance of the fruit counting method proposed.

### 3.2. Performance of the fruit counting algorithm

The counting method developed in this study was tested using the ID-switched number of fruits, MIDE, and RMSE to assess its performance in terms of the matching fruit objects in a video frame. These testing videos were acquired from a commercial orchard near Prosser, WA, USA by controlling the remote control vehicle equipped with a Kinect V2 sensor. One of a sample video (V12) could be found online at https://github.com/fu3lab/Scifresh-apple-images-and-counting-video/blob/main/V12-Fruit-Tracking-Count.mp4). The V12 contains a total of 48 frames, of which 40 frames do not have ID-switched fruits. Table 2 shows the counting results of the remaining 8 frames. Fig. 7 shows 8 consecutive video frames starting from the 5th frame of the V12. The IDs of more than one fruit in the 5th, 12th, and 13th frames of the video V12 were switched, which was caused by the tracking deviation. However, the 8th frame with the largest tracking deviation did not result in fruit ID-switched. This is because when there was a large deviation in the prediction area (OV is less than 40%), the proposed method will automatically adjust the tracking area. There was only one fruit ID-switched in multiple video frames (6th, 18th, 29th, 34th, and 36th frames), which was caused by not all overlapping fruits being detected, as shown in Fig. 8a. The switching of fruit ID is the cause of the MIDE between video frames. The $N_c$ is greater than the $N_g$ in some video frames because the algorithm can detect the fruits that are occluded severely or located on the boundary in the frame, which is also one of the reasons for the emergence of RMSE. Based on Eq. (8) and Eq. (9), the relevant MIDE and RMSE are calculated to be 0.028 and 5.224, respectively, which indicates the reliability of the proposed method in short-term video fruit counting.

**Table 2**
The counting results of 8 frames with ID-switched fruits in the video V12.

| Indicators | Frame | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 5th | 6th | 12th | 13th | 18th | 29th | 34th | 36th |
| $N_s$ | 8 | 1 | 6 | 9 | 1 | 1 | 1 | 1 |
| $N_c$ | 45 | 40 | 17 | 14 | 18 | 45 | 37 | 35 |
| $N_g$ | 53 | 46 | 21 | 18 | 25 | 42 | 42 | 32 |

The counting method shows an average counting accuracy of over 90% on all the 20 orchard videos of varying lengths according to ground truth of visual manual counts. Fig. 9 shows the overall performance of the method presenting the results of linear regression. Each dot in the chart represents the maximum number of the fruit ID obtained in the last frame of a video found by the proposed method in x-axis ($P_t$), and the summation of the visual manual counts in the same video found by inspectors in y-axis ($N_t$). An $R^2$ value of 0.9875 in Fig. 9 suggests that the regression line fits well over the data, which means the computed count of the fruits is similar to the ground truth count. Besides, the number of frames of these videos is between 10 and 104, which contains 68 ~ 494 fruits. It can be concluded that the counting method performed well regardless of the number of video frames. Moreover, $P_c$ values of the above orchard videos are all calculated, which reaches a high average value of 91.49%. It can be seen from Fig. 9 that the minimum $P_c$ value of the proposed method is also higher than 80%. The high counting accuracy of the video also shows the proposed method has superior performance.

FN detection is the opposite of the impact of FP detection and ID-switch on fruit count, which makes it possible to obtain a high counting accuracy. For the selected videos, an average of 37 fruits can be detected per frame, of which about 0.016 FP and 0.098 FN. The FP detections result in an over-count while the FN detections result in an under-count. Failure to detect the fruit that has been assigned an ID in the subsequent frames of the video will cause the fruit to fail to match

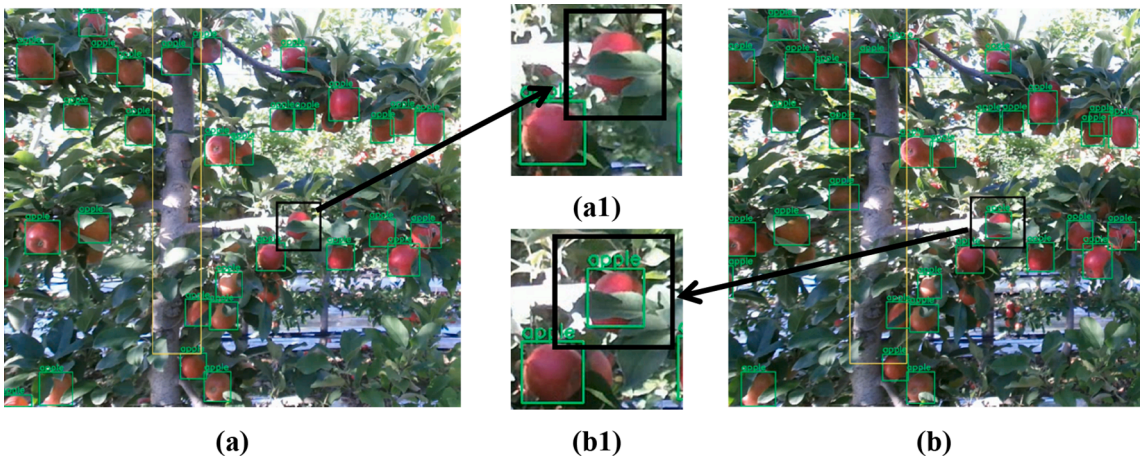
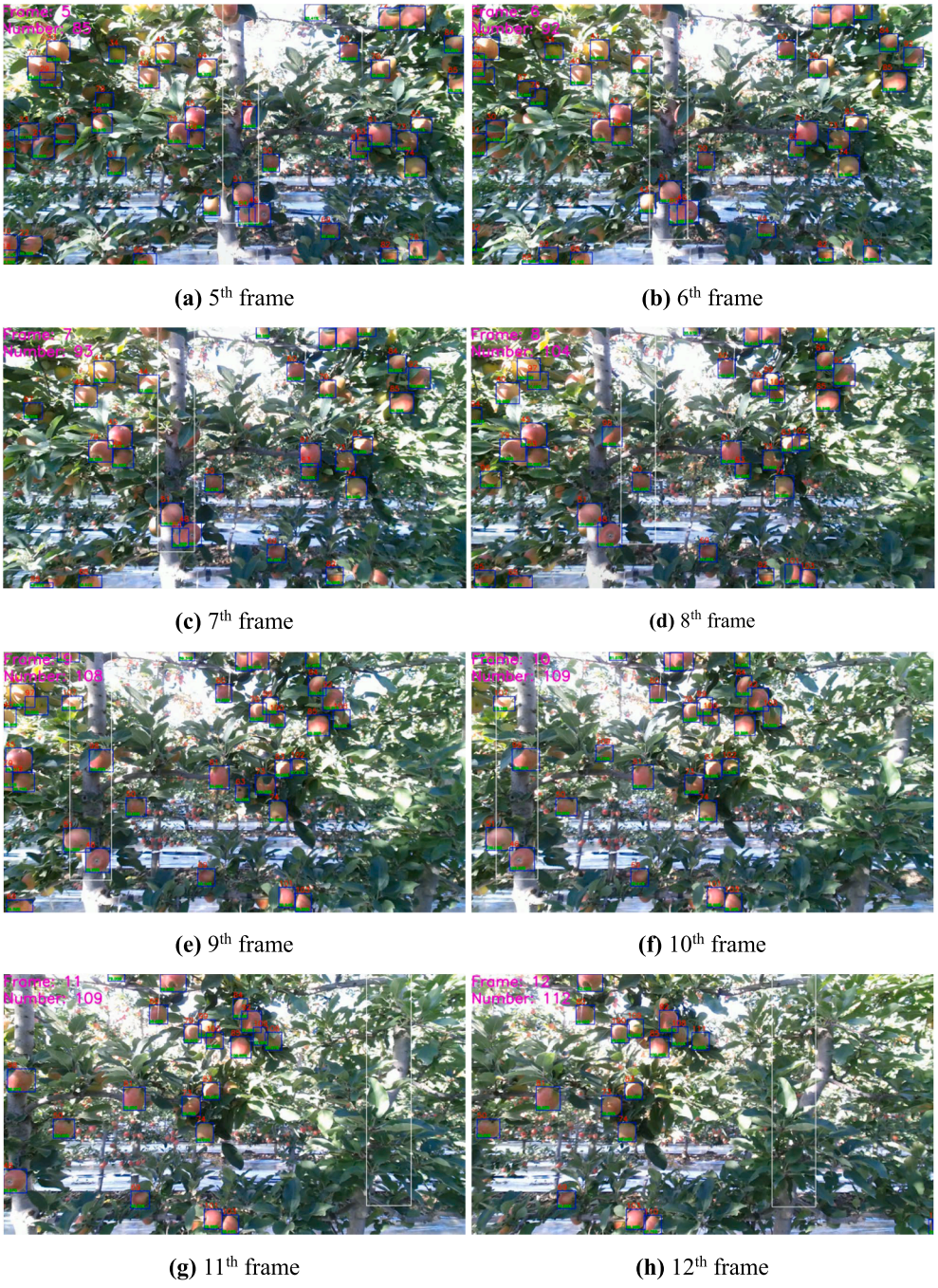



**Fig. 6.** Examples of fruit detection in two successive frames of the video, where the detected fruits and trunks are marked with green and yellow rectangles respectively. **(a)** Fruits detected in $i^{th}$ frame of a video; **(b)** Fruits detected in $(i + 1)^{th}$ frame of the video. The same fruits in the two video frames are marked with black rectangles, and are shown enlarged in **(a1)** and **(b1)**. The fruit is not detected in the $i^{th}$ frame, but is detected in the $(i + 1)^{th}$ frame. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**(a)** 5th frame **(b)** 6th frame

**(c)** 7th frame **(d)** 8th frame

**(e)** 9th frame **(f)** 10th frame

**(g)** 11th frame **(h)** 12th frame

**Fig. 7.** Example of 8 consecutive video frames starting from the 5th frame of the video V12, in which most fruit IDs remain unchanged. The numbers of video frames and fruits are displayed in the upper left corner; All detected fruits are marked by a blue rectangle; The number above the rectangle represents the fruit ID. The traced trunk is inside the white rectangle. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**(a)** No. 58 fruit ID switched **(b)** No. 197 fruit ID switched

**Fig. 8.** Examples of the switch of fruit ID. **(a)** Failure to successfully detect all overlapping fruits causes the fruit ID to switch; **(b)** Failure to continuously successfully detect fruits between video frames causes the fruit ID to switch.

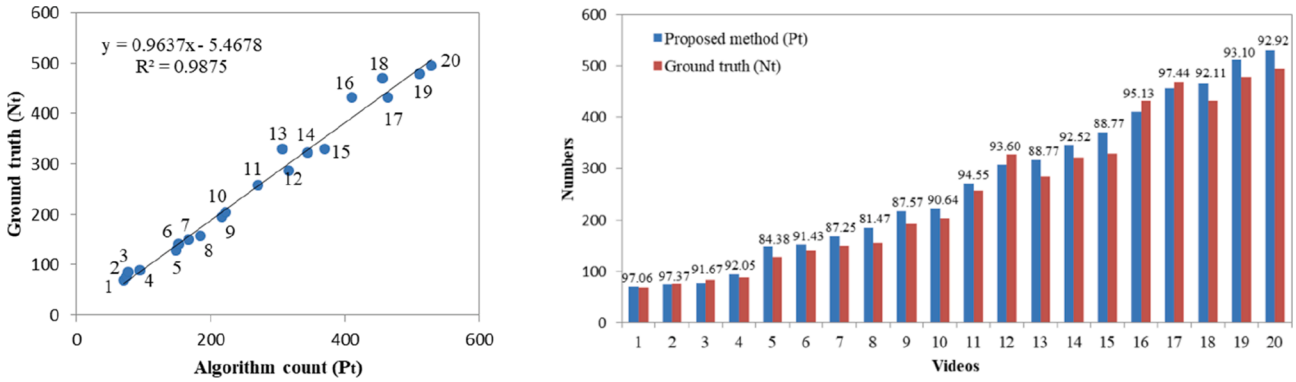



**Fig. 9.** Fruit count results. On the left is a scatter plot of the Proposed method count against Ground truth count. The proposed method counts the 20 videos against the ground truth counts and corresponding $P_c$ values are shown on the right.

the corresponding target in the current frame, resulting in the fruit ID being deleted. Therefore, the fruit will be given another ID when it is successfully detected again, causing the ID of the fruit to switch (Fig. 8b). For the selected video, about 0.094 fruits are switched ID for each video. Fruit ID-switched in the video also results in an over-count. The numbers of over-count fruits and under-count fruits are similar, resulting in a low bias and a high $P_c$ value.

### 3.3. Comparison with other methods

Although there are some reports of apple fruit counts with good results (Roy et al., 2019; Vasconez et al., 2020) it is difficult to compare with this research because the data processed by each research were all captured images or videos from different orchards, and even the quality of captured orchard videos varies greatly. Therefore, an apple fruit counting method based on Deep Sort, a popular multi-object tracking algorithm, is developed to compare the performance of the method proposed in this research. The basic idea of the Deep Sort algorithm is tracking-by-detection. The core is the recursive Kalman filter and the Hungarian algorithm, which employs a metric that combines motion and appearance information to perform data association frame by frame. The algorithm detects the target in each frame of the video and realizes the counting of the target by matching the detected target and the motion trajectory. To better compare the counting performance of the Deep Sort and the method proposed in this research, the YOLOv4-tiny network trained in this research performs the detection work of the Deep Sort algorithm.

Same videos are applied to compare the performance of the two methods, which are shown in Fig. 10a. The proposed method has an average counting accuracy of 91.60% for 5 random videos, while the Deep Sort algorithm has serious over-counting. Although the Deep Sort algorithm has shown good tracking and counting performance for pedestrians, it is not ideal for apples. Because the fruit is not only much smaller than pedestrians or vehicles, there is even no obvious difference in appearance between individuals. Besides, the video obtained in this study was obtained by artificially controlling the remote control vehicle. Although the researcher consciously controlled the average speed of the vehicle, they still obtained videos with a large change in speed, making the movement in the orchard video basically irregular. This not only makes the metric that combines motion and appearance information basically unable to provide an effective function of fruit counting, but also leads to the failure of fruit tracking, as shown in Fig. 10b. In terms of speed, the method proposed in this study can be implemented on CPU at 2 ~ 5 fps, which is faster than Deep Sort whose speed is less than 2 fps.

### 3.4. Future works

The counting method based on trunk tracking proposed in this paper achieved a counting accuracy of more than 90%, but only considered the single-sided row of the tree. Part of the fruits in modern orchard with vertical fruiting-wall architecture can be seen on both sides of the tree row, which results in double counting of those fruits when calculating the number of fruits in the tree row. Therefore, in the future, the study will count the fruits on both sides of the tree row by removing the fruits that are visible on both sides to accurately estimate the orchard yield. Although there are some studies to reduce the existence of this error by establishing mathematical models (Chen et al., 2017; Koirala et al., 2019), it is difficult for them to cope with sudden changes in orchard

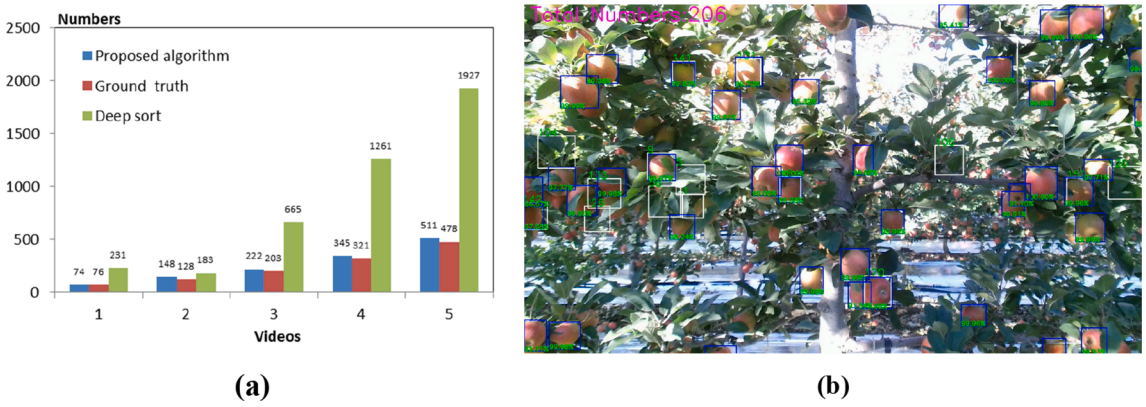



**Fig. 10.** **(a)** Comparison of fruit counting results based on 5 videos; **(b)** Example of tracking failure of Deep Sort algorithm. The rectangles of blue and white colors are referring to the detected apple fruits and tracked fruits, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

yields. Roy et al. (2018) proposed a method for detecting the visible fruits on both sides of a tree row based on three-dimensional reconstruction technology, which can help to count fruits. Dong et al. (2020) applied the method proposed by Roy et al. (2018) to estimate the yield of apple orchards, but showed the method estimated that a row of trees requires more than 34 min of processing time. Therefore, methodologies that can identify the fruits visible on both sides of the tree row, combined with the current fruit counting algorithm to obtain the number of fruits in the orchard tree row in real-time, is the main future work.

## 4. Conclusions

This research examined automatically counting apple fruits (during harvest season) on vertical-fruiting-wall trees for estimation orchard yield using a method developed by YOLOv4-tiny and CSR-DCF-based tracker in orchard videos. The proposed method mainly includes detection, tracking, and counting. To count fruits, both the fruit and trunk were selected as detection objects, while the trunk was selected as the tracking object because of its obvious characteristics and large volume. This research found that the fruits in the video can be successfully detected and counted based on deep learning and trunk tracking. The network trained using YOLOv4-tiny reached the AP values of 99.59% and 99.10% for fruits and trunks, respectively. Besides, the network only took 0.022 s on average to process an orchard image with a resolution of 1920 × 1080 pixels. This high AP and processing speed guarantee the successful run of the counting method. Tracking the trunk by establishing CSR-DCF-based trackers obtained a MIDE of 0.028 and an RMSE of 5.224 for fruits, which showed the reliability of the tracker. An $R^2$ value of 0.9875 and a high average counting accuracy of 91.49% were achieved in the orchard videos. The counting performance was better than the counting method developed based on a multi-object tracking algorithm (Deep Sort). Moreover, the proposed method can be implemented on CPU at 2 ~ 5 fps, which provides the possibility for real-time yield estimation of the orchard. However, this study only considered the single-sided row of the tree when counting, resulting in double counting of fruits that appeared on both sides of the tree row. Therefore, methods of removing fruits visible on both sides of tree rows will be investigated in the future to accurately estimate orchard yields.

## CRediT authorship contribution statement

**Fangfang Gao:** Data curation, Investigation, Methodology, Writing – original draft. **Wentai Fang:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Xiaoming Sun:** Conceptualization, Methodology, Writing – review & editing. **Zhenchao Wu:** Conceptualization, Methodology, Writing – review & editing. **Guanao Zhao:** Investigation, Conceptualization, Writing – review & editing. **Guo Li:** Investigation, Methodology, Writing – review & editing. **Rui Li:** Methodology, Supervision, Writing – review & editing. **Longsheng Fu:** Conceptualization, Data curation, Methodology, Supervision, Writing – review & editing. **Qin Zhang:** Methodology, Writing – review & editing.

## Declaration of Competing Interest

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

## Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (32171897); Youth Science and Technology Nova Program in Shaanxi Province of China (2021KJXX-94); Science and Technology Promotion Program of Northwest A&F University (NWAFU, TGZX2021-29); China Postdoctoral Science Foundation funded project (2019M663832); Recruitment Program of High-End Foreign Experts of the State Administration of Foreign Experts Affairs, Ministry of Science and Technology, China (G20200027075), and Washington State University (WSU) Center for Precision and Automated Agricultural Systems. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the view of the funding agencies, NWAFU and WSU.

## References

Apolo-Apolo, O.E., Martínez-Guanter, J., Egea, G., Raja, P., 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 https://doi.org/10.1016/j.eja.2020.126030.

Bargoti, S., Underwood, J.P., 2017. Image segmentation for fruit detection and yield estimation in apple orchards. J. Field Rob. 34 (6), 1039–1060. https://doi.org/10.1002/rob.21699.

Chen, S.W., Shivakumar, S.S., Dcunha, S., Das, J., Okon, E., Qu, C., Taylor, C.J., Kumar, V., 2017. Counting apples and oranges with deep learning: A data-driven approach. IEEE Rob. Autom. Lett. 2 (2), 781–788. https://doi.org/10.1109/LRA.2017.2651944.

Danelljan, M., Häger, G., Khan, F.S., Felsberg, M., 2014. In: Accurate scale estimation for robust visual tracking. British Machine Vision Association, Worcester, England. https://doi.org/10.5244/C.28.65.

Dong, W., Roy, P., Isler, V., 2020. Semantic mapping for orchard environments by merging two-sides reconstructions of tree rows. J. Field Rob. 37 (1), 97–121. https://doi.org/10.1002/rob.21876.

Dorj, U.O., Lee, M., Yun, S.S., 2017. An yield estimation in citrus orchards via fruit detection and counting using image processing. Comput. Electron. Agric. 140, 103–112. https://doi.org/10.1016/j.compag.2017.05.019.

Farhodov, X., Kwon, O.-H., Moon, K.-S., Kwon, O., Lee, S.-H., Kwon, K.-R., 2019. A new CSR-DCF tracking algorithm based on Faster RCNN detection model and CSRT tracker for drone data. Journal of Korea Multimedia Society 22 (12), 1415–1429. https://doi.org/10.9717/kmms.2019.22.12.1415.

Fountas, S., Espejo-Garcia, B., Kasimati, A., Mylonas, N., Darra, N., 2020. The future of digital agriculture: Technologies and opportunities. IT Prof. 22 (1), 24–28. https://doi.org/10.1109/MITP.2019.2963412.

Fu, L., Feng, Y., Wu, J., Liu, Z., Gao, F., Majeed, Y., Al-Mallahi, A., Zhang, Q., Li, R., Cui, Y., 2021. Fast and accurate detection of kiwifruit in orchard using improved YOLOv3-tiny model. Precis. Agric. 22 (3), 754–776. https://doi.org/10.1007/s11119-020-09754-y.

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

Fu, L., Majeed, Y., Zhang, X., Karkee, M., 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. https://doi.org/10.1016/j.biosystemseng.2020.07.007.

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

Gao, F., Wu, Z., Suo, R., Zhou, Z., Li, R., Fu, L., Zhang, Z., 2021. Apple detection and counting using real-time video based on deep learning and object tracking. Trans. Chinese Soc. Agric. Eng. 37, 217–224. https://doi.org/10.11975/j.issn.1002-6819.2021.21.025.

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

Grilli, E., Battisti, R., Remondino, F., 2021. An advanced photogrammetric solution to measure apples. Remote Sensing 13 (19), 3960. https://doi.org/10.3390/rs13193960.

He, L., Fang, W., Zhao, G., Wu, Z., Fu, L., Li, R., et al., 2022. Fruit yield prediction and estimation in orchards: A state-of-the-art comprehensive review for both direct and indirect methods. Comput. Electron. Agric. 195, 106812. https://doi.org/10.1016/j.compag.2022.106812.

Jain, M., Gemert, J.V., Bouthemy, P., Snoek, C.G.M., 2014. In: Action localization with tubelets from motion. IEEE Computer Society, pp. 740–747. https://doi.org/10.1109/CVPR.2014.100.

Jiao, Y., Luo, R., Li, Q., Deng, X., Yin, X., Ruan, C., Jia, W., 2020. Detection and localization of overlapped fruits application in an apple harvesting robot. Electronics 9 (6), 1–14. https://doi.org/10.3390/electronics9061023.

Kalal, Z., Mikolajczyk, K., Matas, J., 2012. Tracking-learning-detection. IEEE Trans. Pattern Anal. Mach. Intell. 34 (7), 1409–1422. https://doi.org/10.1109/TPAMI.2011.239.

Koirala, A., Walsh, K.B., Wang, Z., McCarthy, C.L., 2019. Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of 'MangoYOLO'. Precis. Agric. 20 (6), 1107–1135. https://doi.org/10.1007/s11119-019-09642-0.

Kuznetsova, A., Maleva, T., Soloviev, V., 2020. Using YOLOv3 algorithm with pre- And post-processing for apple detection in fruit-harvesting robot. Agronomy 10 (7), 1016. https://doi.org/10.3390/agronomy10071016.

Lee, H., Cho, A., Lee, S., Whang, M., 2019. Vision-based measurement of heart rate from ballistocardiographic head movements using unsupervised clustering. Sensors 19, 3263. https://doi.org/10.1201/9781315382586-12.

Li, B., Wu, T., Zhu, S.C., 2014. Integrating context and occlusion for car detection by hierarchical and-or model. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 8694, 652–667. https://doi.org/10.1007/978-3-319-10599-4_42.

Li, G., Suo, R., Zhao, G., Gao, C., Fu, L., Shi, F., et al., 2022. Real-time detection of kiwifruit flower and bud simultaneously in orchard using YOLOv4 for robotic pollination. Comput. Electron. Agric. 193, 106641. https://doi.org/10.1016/j.compag.2021.106641.

Li, Y., Feng, X., Liu, Y., Han, X., 2021. Apple quality identification and classification by image processing based on convolutional neural networks. Sci. Rep. 11 (1), 1–15. https://doi.org/10.1038/s41598-021-96103-2.

Lin, G., Tang, Y., Zou, X., Xiong, J., Fang, Y., 2020. Color-, depth-, and shape-based 3D fruit detection. Precis. Agric. 21 (1), 1–17. https://doi.org/10.1007/s11119-019-09654-w.

Lin, T., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., et al., 2015. In: Microsoft COCO: Common objects in context. Springer Verlag, Berlin, Germany. https://doi.org/10.1007/978-3-319-10602-1_48.

Liu, X.u., Chen, S.W., Liu, C., Shivakumar, S.S., Das, J., Taylor, C.J., Underwood, J., Kumar, V., 2019a. Monocular camera based fruit counting and mapping with semantic data association. IEEE Rob. Autom. Lett. 4 (3), 2296–2303.

Liu, X., Zhao, D., Jia, W., Ji, W., Sun, Y., 2019b. A detection method for apple fruits based on color and shape features. IEEE Access 7, 67923–67933. https://doi.org/10.1109/ACCESS.2019.2918313.

Lu, S., Chen, W., Zhang, X., Karkee, M., 2022. Canopy-attention-YOLOv4-based immature/mature apple fruit detection on dense-foliage tree architectures for early crop load estimation. Comput. Electron. Agric. 193, 106696 https://doi.org/10.1016/j.compag.2022.106696.

Lukežič, A., Vojíř, T., Čehovin Zajc, L., Matas, J., Kristan, M., 2018. Discriminative Correlation Filter Tracker with Channel and Spatial Reliability. Int. J. Comput. Vision 126 (7), 671–688. https://doi.org/10.1007/s11263-017-1061-3.

Milan, A., Rezatofighi, S.H., Dick, A., Reid, I., Schindler, K., 2017. In: Online multi-target tracking using recurrent neural networks. AAAI Press, pp. 4225–4232.

Montalbo, F.J.P., 2020. A computer-aided diagnosis of brain tumors using a fine-tuned yolo-based model with transfer learning. KSII Trans. Internet Inf. Syst. 14 (12), 4816–4834. https://doi.org/10.3837/tiis.2020.12.011.

Ngo, T.N., Wu, K.C., Yang, E.C., Lin, T.T., 2019. A real-time imaging system for multiple honey bee tracking and activity monitoring. Comput. Electron. Agric. 163, 104841 https://doi.org/10.1016/j.compag.2019.05.050.

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

Payne, A., Walsh, K., Subedi, P., Jarvis, D., 2014. Estimating mango crop yield using image analysis using fruit at "stone hardening" stage and night time imaging. Comput. Electron. Agric. 100, 160–167. https://doi.org/10.1016/j.compag.2013.11.011.

Qureshi, W.S., Payne, A., Walsh, K.B., Linker, R., Cohen, O., Dailey, M.N., 2017. Machine vision for counting fruit on mango tree canopies. Precis. Agric. 18 (2), 224–244. https://doi.org/10.1007/s11119-016-9458-5.

Roy, P., Dong, W., Isler, V., 2018. In: Registering reconstructions of the two sides of fruit tree rows. Institute of Electrical and Electronics Engineers Inc., New York, USA, pp. 7697–7702. https://doi.org/10.1109/IROS.2018.8594167.

Roy, P., Kislay, A., Plonski, P.A., Luby, J., Isler, V., 2019. Vision-based preharvest yield mapping for apple orchards. Comput. Electron. Agric. 164, 104897 https://doi.org/10.1016/j.compag.2019.104897.

Song, Z., Zhou, Z., Wang, W., Gao, F., Fu, L., Li, R., Cui, Y., 2021. Canopy segmentation and wire reconstruction for kiwifruit robotic harvesting. Comput. Electron. Agric. 181, 105933 https://doi.org/10.1016/j.compag.2020.105933.

Stein, M., Bargoti, S., Underwood, J., 2016. Image based mango fruit detection, localisation and yield estimation using multiple view geometry. Sensors 16 (11), 1915. https://doi.org/10.3390/s16111915.

Suo, R., Gao, F., Zhou, Z., Fu, L., Song, Z., Dhupia, J., et al., 2021. Improved multi-classes kiwifruit detection in orchard to avoid collisions during robotic picking. Comput. Electron. Agric. 182, 106052 https://doi.org/10.1016/j.compag.2021.106052.

Vasconez, J.P., Delpiano, J., Vougioukas, S., Auat Cheein, F., 2020. Comparison of convolutional neural networks in fruit detection and counting: A comprehensive evaluation. Comput. Electron. Agric. 173, 105348 https://doi.org/10.1016/j.compag.2020.105348.

Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y. M., 2020. Scaled-YOLOv4: Scaling Cross Stage Partial Network. http://arxiv.org/abs/2011.08036 (accessed 10 July 2020).

Wang, C., Lee, W.S., Zou, X., Choi, D., Gan, H., Diamond, J., 2018. Detection and counting of immature green citrus fruit based on the Local Binary Patterns (LBP) feature using illumination-normalized images. Precis. Agric. 19 (6), 1062–1083.

Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.S., 2019a. In: Fast online object tracking and segmentation: A unifying approach. IEEE Computer Society, pp. 1328–1338. https://doi.org/10.1109/CVPR.2019.00142.

Wang, X., Kang, H., Zhou, H., Au, W., Chen, C., 2022. Geometry-aware fruit grasping estimation for robotic harvesting in apple orchards. Comput. Electron. Agric. 193, 106716 https://doi.org/10.1016/j.compag.2022.106716.

Wang, Z., Walsh, K., Koirala, A., 2019b. Mango fruit load estimation using a video based MangoYOLO—Kalman filter—hungarian algorithm method. Sensors 19 (12), 2742. https://doi.org/10.3390/s19122742.

Wawrzyniak, N., Hyla, T., Popik, A., 2019. Vessel detection and tracking method based on video surveillance. Sensors 19 (23), 5230. https://doi.org/10.3390/s19235230.

Wojke, N., Bewley, A., Paulus, D., 2018. Simple online and realtime tracking with a deep association metric. In: 24th IEEE International Conference on Image Processing, pp. 3645–3649. https://doi.org/10.1109/ICIP.2017.8296962.

Wu, Z., Li, G., Yang, R., Fu, L., Li, R., Wang, S., 2022. Coefficient of restitution of kiwifruit without external interference. J. Food Eng. 327, 111060 https://doi.org/10.1016/j.jfoodeng.2022.111060.

Wu, Z., Yang, R., Gao, F., Wang, W., Fu, L., Li, R., 2021. Segmentation of abnormal leaves of hydroponic lettuce based on DeepLabV3+ for robotic sorting. Comput. Electron. Agric. 190, 106443 https://doi.org/10.1016/j.compag.2021.106443.

Zhang, Z., Flores, P., Igathinathane, C., Naik, D.L., Kiran, R., Ransom, J.K., 2020. Wheat lodging detection from UAS imagery using machine learning algorithms. Remote Sensing 12, 1838. https://doi.org/10.3390/rs12111838.

Zhou, Z., Song, Z., Fu, L., Gao, F., Li, R., Cui, Y., 2020. Real-time kiwifruit detection in orchard using deep learning on Android™ smartphones for yield estimation. Comput. Electron. Agric. 179, 105856 https://doi.org/10.1016/j.compag.2020.105856.