BIOSYSTEMS ENGINEERING 234 (2023) 140–155

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/issn/15375110

Research Paper

Twice matched fruit counting system: An automatic fruit counting pipeline in modern apple orchard using mutual and secondary matches

Zhenchao Wu [a], Xiaoming Sun [a], Hanhui Jiang [a], Fangfang Gao [a], Rui Li [a,d], Longsheng Fu [a,b,c,d,*], Dong Zhang [e], Spyros Fountas [f]

[a] College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling, Shaanxi, 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] College of Horticulture, Northwest A&F University, Yangling, Shaanxi, 712100, China
[e] Northwest A&F University Shenzhen Research Institute, Shenzhen, Guangdong 518000, China
[f] Agricultural University of Athens, Athens 11855, Greece

ARTICLE INFO

Article history:
Received 3 March 2023
Received in revised form
31 August 2023
Accepted 11 September 2023
Published online 20 September 2023

Keywords:
Yield estimation
Clustered fruit
Object detection
Mutual match
ID assignment

Fruit counting, as one of the essential parts of yield estimation, is an important factor in production process planning. In the case of apple crops, it is useful in orchard management and as guidance for farmers, showing a decisive role in product market strategies and cultivation practices. Although some machine vision based studies have exhibited notable fruit counting ability, they still need to be improved for clustered fruit. This study proposes an automatic fruit counting pipeline called twice matched fruit counting system to overcome this limitation. The twice matched fruit counting system consists of three sub-algorithms: i) object detection model based on You Only Look Once Version 4-tiny; ii) fruit tracking with mutual match; iii) and fruit counting with ID assignment. The object detection model was developed based on You Only Look Once Version 4-tiny, which quickly and accurately detect fruit and trunks with mean average precision of 96.4% and detection speed of 16 ms. The fruit tracking with mutual match was designed to alleviate match errors associated with the clustered fruit, which achieved superior performance with average ID Switch Rate of 3.9%, Multiple Object Tracking Accuracy of 89.9% and Multiple Object Tracking Precision of 93.5%. The fruit counting was implemented by ID assignment, where each fruit was assigned with a unique ID based on fruit tracking results and direction of camera motion. The root mean squared error and coefficient of determination were 16.3 fruit per video and 0.93, respectively, which indicate a high correlation between fruit count results from the proposed approach and ground truth counting results. The twice matched fruit counting system was implemented on Central Processing Unit at

* *Corresponding author*. College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling, Shaanxi, 712100, China.
E-mail address: fulsh@nwafu.edu.cn (L. Fu).
https://doi.org/10.1016/j.biosystemseng.2023.09.005

3–5 frames per second. These results demonstrate a potential of the twice matched fruit counting system for estimating fruit yield in modern apple orchards, which could provide technical support for orchard management.

Nomenclature

Symbols

AP	Average precision of object detection model, %
AP_{fruit}	Average precision of fruit detection model, %
AP_{trunk}	Average precision of trunk detection model, %
mAP	Mean average precision of object detection model, %
P	Precision of object detection model, %
R	Recall of object detection model, %
W_{ID}	ID Switch Rate, %
P_{tr}	Multiple Object Tracking Accuracy, %
P_{mt}	Multiple Object Tracking Precision, %
P_c	Counting accuracy, %
(x^i, y^i)	The coordinates of the centre position of detected object Bounding boxes for video frame i, pixel
$RMSE$	Root mean squared error, fruit per video
R^2	Coefficient of determination
D_{dev}	The variation in the bounding box size of the fruit from a YOLO detector between consecutive frames, pixel
Q_{sh}	The number of fruit with ID switching
S	The number of algorithm counted fruit
M_{mat}	The number of correctly tracked fruit
T_{mat}	Total number of matched fruit
G_{manu}	The number of ground truth
N_{count}	Fruit count results for the current frame
TP	True positive, means the number of correctly detected objects
FN	False negative, means the number of missed or undetected objects
FP	False positive, means the number of falsely detected objects

Abbreviations

RGB	Red, Green, and Blue
CPU	Central Processing Unit
YOLO	You Only Look Once
V4-tiny	Version 4-tiny
V3	Version 3
SGD	Stochastic gradient descent
IoU	Intersection over Union

1. Introduction

Yield estimation enables growers to obtain information about orchards to formulate optimal orchard management. It also helps growers in decision-making related to required human resources, harvesting and storage facilities, transportation, and product marketing (Ma et al., 2021; Meng et al., 2020; Sheng et al., 2021; Zhang et al., 2021). However, current practice for yield estimation typically relies only on cursory observations, which are time-consuming and less reliable (Qureshi et al., 2017). With the introduction of computer vision in the smart agriculture domain, automated yield estimation methodologies have been extensively researched to address this problem (Dorj et al., 2017; Ni et al., 2020; Rahnemoonfar & Sheppard, 2017; Stein et al., 2016; Yang & Xu, 2021). These computer vision techniques resulted in improved yield estimation accuracy in less time.

Fruit counting is the fundamental part of automated yield estimation, which mainly relies on detecting and counting fruit in images. Mekhalfi et al. (2020) and Massah et al. (2021) applied Viola–Jones object detection algorithm and support vector machine, respectively, to predict the number of kiwifruits in an image. Behera et al. (2021) and Koirala et al. (2019) developed mango detection networks based on Faster RCNN and YOLO, respectively, to count the number of fruit in an image. There are various studies on the detection and counting of citrus fruit (Apolo-Apolo et al., 2020; Chen et al., 2017; Dorj et al., 2017; Maldonado & Barbosa, 2016; Wang et al., 2018). The above mentioned studies have reached the correction coefficient of 0.93 for fruit counting. These encouraging results have laid a foundation for counting studies on other types of fruit or crops. Liu et al. (2019) highlighted three issues related to fruit counting in modern apple orchard: (1) double counting the same fruit in consecutive images; (2) double counting the same fruit from both sides of the tree; and (3) double counting fruit that are initially tracked, then lost, and then detected and tracked again in a later image.

Fruit counting in modern apple orchard has been achieved by tracking fruit from tree row videos, which is the key to avoid double counting the same fruit detected in consecutive images. Various tracking algorithms have been developed for pedestrians (Al-Sa'd et al., 2022; Wang et al., 2022) or cars (Badue et al., 2021; Jiang et al., 2022) on the road. However, these tracking algorithms often do not perform well for fruit tracking in orchards, because individual characteristics of fruit are similar and difficult to distinguish (Zhang et al., 2022). Modern apple orchards have narrow canopies making fruit visible from the inter-row, which allows tree row videos to be used for fruit counting purpose. Vasconez et al. (2020) used multi-object tracking based on Bayesian filter algorithm to count fruit in the video with counting accuracy of 93.0%. Wang et al. (2019) innovatively introduced a 'borrow' concept to predict fruit position by borrowing the speed vector of neighbouring fruit, which resulted in an estimate of 'non-hidden' fruit that is only 2.6% more than the count of the harvest tally. Roy et al. (2019) achieved a counting accuracy of

more than 89.0% by tracking fruit in multiple video frames based on pairwise homography modelled camera motion, which was estimated by matching Scale Invariant Feature Transform (SIFT) features above the ground. However, the algorithm for direct tracking of fruit has a high computing resource requirement, which makes it difficult to ensure real-time performance. Considering trunk (which is stationary object relative to fruit on tree) as tracking object to calculate displacements of the video motion for counting fruit, Gao et al. (2022) resulted in counting accuracy of 91.5% for tree row videos at 2–5 frames per second (fps) on Central Processing Unit (CPU).

Although a correlation filter-based on trunk tracking method has been developed to improve the speed of fruit counting, it is only applicable to modern apple orchards with vertical fruiting-wall architecture. In vertical fruiting-wall orchard architecture, tree trunks are straight and have distinguish features which could be accurately tracked based on the correlation filter (Lukežič et al., 2018). However, in other densely planted orchards with dwarf crowns, trunks are often not straight due to a large number of fruit and lack of wire restraints, making it difficult for the correlation filter to accurately predict the trunk area. Most of the modern apple orchards in China are densely planted with dwarf crowns. Therefore, correlation filter-based on trunk tracking method is not suitable for fruit counting because of the importance of predicting trunk area in calculating displacements of the video motion. Gao et al. (2021) showed that the precise area and position of the target can be accurately detected using YOLO series object detection networks. Therefore, this study implements trunk tracking by matching detected trunks with a minimum Euclidean distance in consecutive video frames to calculate the displacement of video motion.

Additionally, clustered fruit is widely distributed in modern apple orchards, which may cause detected fruit to be mismatched with predicted fruit based on the displacement of video motion. Gao et al. (2021) achieved relatively lower fruit counting accuracy of 81.9% due to the mismatch caused by the clustered fruit in orchards. Unlike vehicles or pedestrians, the similarity in appearance between fruit makes it difficult to abate this mismatch (Wang et al., 2022). Hence, the developed fruit counting algorithms should be able to correctly match the clustered fruit in consecutive video frames to ensure the accuracy of counting.

In this study, an automatic fruit counting pipeline named as twice matched fruit counting system was developed, which includes three sub-algorithms: i) object detection model based on You Only Look Once Version 4-tiny (YOLOv4-tiny) to detect fruit and trunk; ii) fruit tracking with mutual match; and iii) fruit counting with ID assignment. In the first stage, YOLOv4-tiny was used to detect fruit and trunks. Then, fruit tracking with mutual match was developed based on the displacement of video motion (which was computed by tracking the trunk detected in consecutive video frames). At the end, fruit were assigned with IDs for counting.

2. Materials and methods

This study aims to develop an automatic fruit counting system for yield estimation in modern apple orchards. The proposed pipeline of the twice matched fruit counting system is given in Fig. 1. Firstly, images and videos of tree row were collected in a modern apple orchard to build datasets. Secondly, an appropriate YOLO network was selected to detect fruit and trunks. Thirdly, the displacement of video motion was calculated to track fruit in consecutive video frames based on detection results. Finally, fruit were assigned with IDs for counting. Details outlining each step are provided below.

2.1. Dataset preparation

Dataset preparation of modern apple orchard is the basis for fruit counting. By processing RGB (Red, Green, and Blue) images and videos of tree row collected in experimental orchard, datasets for target detection, tracking and counting were created. The specific process is shown in Fig. 2. Experimental videos contained 10 videos of 135–295 frames, whose corresponding tree rows have lengths of 6.0–10.0 m, named as Vid_1, Vid_2, Vid_3, Vid_4, Vid_5, Vid_6, Vid_7, Vid_8, Vid_9, and Vid_10. Video dataset could be found online at https://github.com/fu3lab/Apple_FruitCounting.

2.1.1. Data acquisition

Dataset for this study was collected from densely planted with dwarf crowns apple orchard located in Famen Town, Baoji City, Shaanxi Province, China (34°29′41″ N latitude, 107°51′40″ E longitude). Inter-plant and inter-row spacings of this orchard were 1.5 m and 4.0 m, respectively. Original videos and images were collected with an Intel RealSense D435 camera mounted on a remotely controlled vehicle, as shown in Fig. 3(a). The length of the imaged tree rows ranged from 20.0 to 30.0 m. A total of 1000 original images and 20 original videos with a resolution of 720 × 1280 pixels (RGB data) were acquired under different lighting conditions including front lighting and back lighting, as shown in Fig. 3(b). This dataset was acquired between 8:00 a.m. and 6:00 p.m. on 30th September 2020 to 25th October 2021. Both acquisitions had the same amount of data and belonged to the mature stage of fruit. Original videos were acquired at 30 fps. The moving speed of the remote-controlled vehicle was about 0.5–1.0 m/s due to the muddy ground, which resulted in objects translating about 10–20 pixels between consecutive frames.

In practical application, much faster speed is required (to assess whole orchards) at (ideally) lower frame rate (to reduce

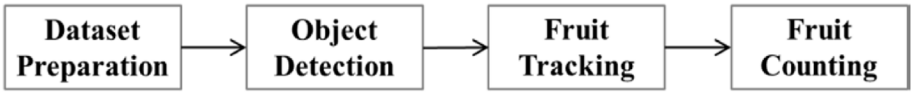


Fig. 1 – Proposed pipeline of the twice matched fruit counting system.

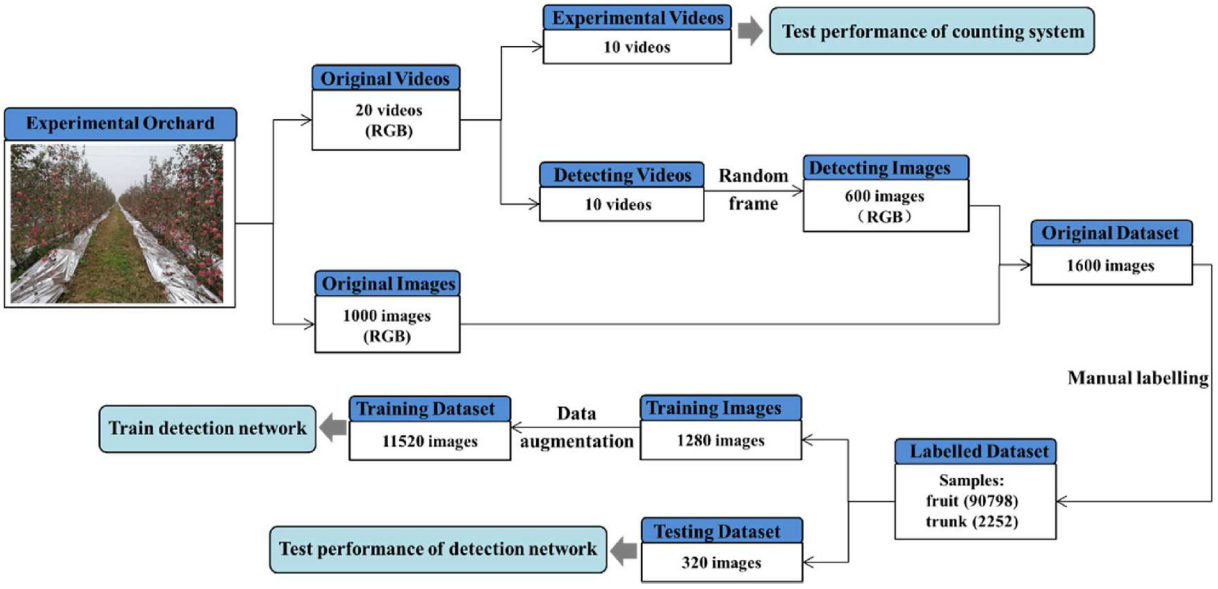


Fig. 2 – Process of data preparation.

Fig. 3 – Schematic of data acquisition in the experimental orchard. The distance between the camera and tree row is 2.0 m; The camera is installed on a tower structure of a remote-controlled vehicle about 1.4 m above the ground. (a) Original images and videos of tree row are obtained by controlling the remote-controlled vehicle. Blue arrow refers to the shooting direction of the camera, while red arrow refers to the moving direction of the remote-controlled vehicle. (b) An example of original images or original videos of tree row in a modern apple orchard. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

computing resource use), meaning more separation between objects in consecutive frames. These were emulated by dropping frames before analysis, i.e., from 30 fps to an effective 20 and 15 fps. Although there may be some immature fruit at the mature stage, their appearance is almost identical to mature fruit and there is no way to distinguish them from mature fruit based on RGB data alone. Therefore, all fruit in original images and videos were considered as mature fruit.

2.1.2. *Dataset building*

An original dataset was constructed using original images and randomly selected frames of original videos to train object detection network. Original videos were equally divided (10 each) into experimental and detecting videos. A total of 600 detecting images were extracted from the detecting videos, together with the 1000 original images captured by the camera forming the original dataset. These images were labelled by

Table 1 – Details of training parameters.

Learning rate	0.001
Weight decay	0.0005
Batch	64
Maximum batches	50,000
Learning method	Stochastic gradient descent (SGD)
Maximum epoch	500
Train crop size	[416, 416]
Momentum	0.9
Framework	Darknet

LabelImg software using rectangular boxes, which was the same as Gao et al. (2022) and took about 400 h. Fruit on the ground and in the back row of trees were not labelled in the labelled dataset, which contained 93,050 instances of fruit and trunks. Data augmentation, including brightness, contrast transformations, Gaussian blur, sharpness transformations, motion blur transformation, and image mirroring in the horizontal axis, was implemented to enlarge the number of training images from 1280 to 11,520.

2.2. Object detection

The twice matched fruit counting system started with frame-by-frame detection of fruit and trunks. This study explores possible object detection networks and makes decisions based on their detection performance.

2.2.1. Network selection

Recent studies revealed that YOLO light network could greatly improve detection speed while ensuring detection accuracy. YOLOv4-tiny was selected because of its high detection accuracy and real-time speed.

2.2.2. Network training

Network training and testing were implemented on a desktop computer with NVIDIA GTX 1080 8 GB GPU, Intel Core i5-6400 (2.70 GHz) CPU, 16 GB of RAM, and 64-bit Windows 10. Software included Python 3.6, Microsoft Visual Studio 2015, cuDNN 7.1.3, CMake-3.16, CUDA 9.0, and OpenCV 3.1.0. Details of training parameters are shown in Table 1. A transfer learning technique was applied to train the network involved training on the COCO dataset (Lin et al., 2014).

2.3. Fruit tracking and counting

Fruit tracking, as a key component of the twice matched fruit counting system, was implemented by matching and associating the same fruit in consecutive video frames based on reference displacement. Fruit tracking strategies were mainly divided into mutual and secondary matches, as shown in Fig. 4. In mutual match, displacement of the same trunk between consecutive video frames was calculated based on detected trunks. Since trunk is stationary relative to fruit on tree, the displacement of the same trunk between consecutive video frames was employed as a reference displacement to predict fruit position. The same fruit between consecutive video frames was matched by mutual matching detected fruit and predicted fruit ($Fruit_{pre}$) to abate mismatch caused by the clustered fruit. Mutual match was performed based on the minimum Euclidean distance between the centre positions of the predicted fruit and the detected fruit bounding boxes in

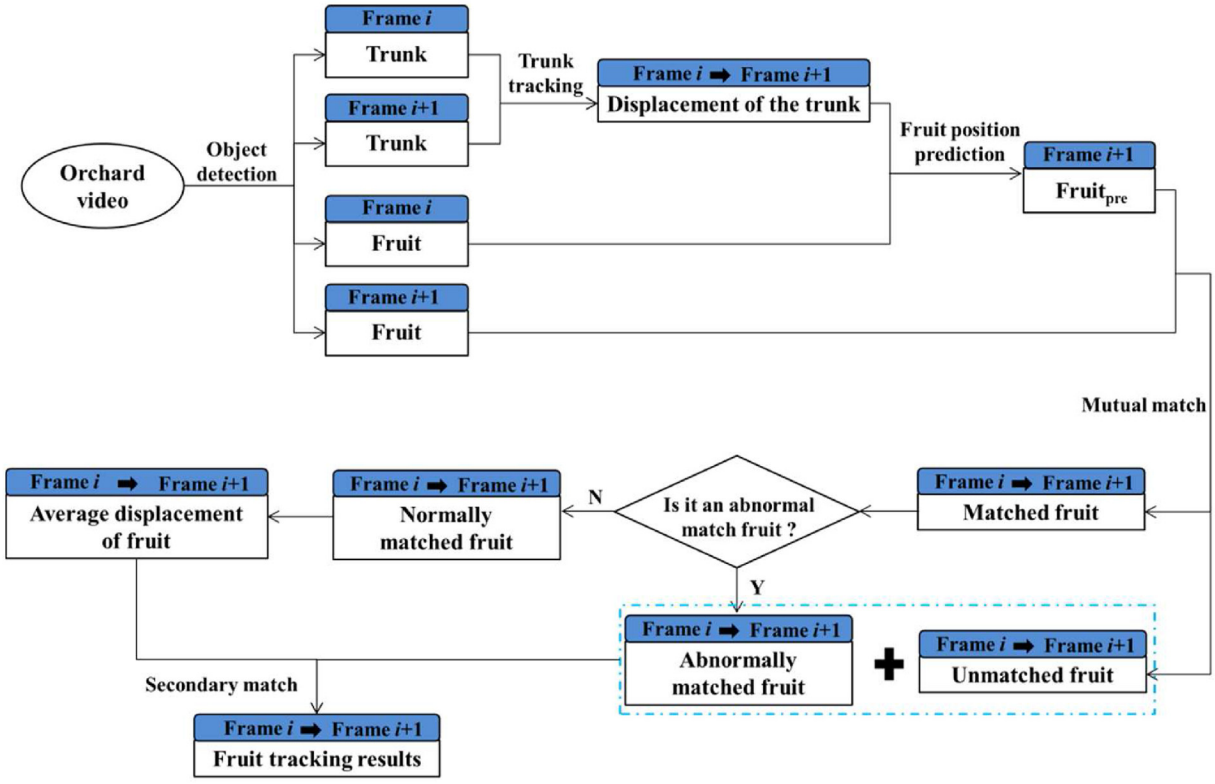
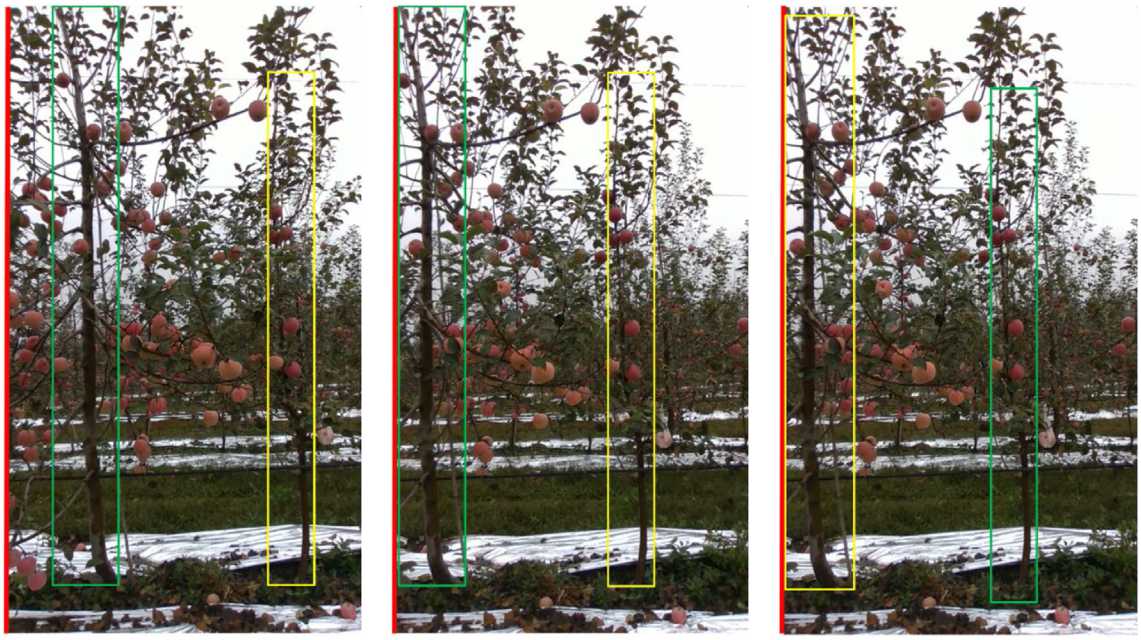


Fig. 4 – Fruit tracking strategies based on mutual and secondary matches.



(a) j^{th} frame **(b)** $(j+17)^{th}$ frame **(c)** $(j+18)^{th}$ frame

Fig. 5 – Transformation of the tracked trunk between consecutive video frames. The left boundary of the video field of view is marked by the solid red line. The detected and tracked trunk is marked with the green bounding box, which provides the reference displacement to predict fruit positions between consecutive video frames. Detected trunks is marked with the yellow bounding box. (a) Trunks that are tracked and detected in the *j*th video frame. (b) and (c) refer to a transformation of the tracked trunk between consecutive frames. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

the current frame. Fruit were divided into matched and unmatched fruit after mutual matching. Matched fruit were divided into normally matched and abnormally matched fruit based on their motion trajectory. In secondary match, displacements of normally matched fruit between the previous frame and the current frame were averaged as a new reference displacement in place of the displacement of the same trunk between consecutive video frames. Unmatched and abnormally matched fruit were secondary matched to obtain the maximum number of normally matched fruit based on Intersection over Union (IoU) between bounding boxes of the predicted fruit and the detected fruit in the current frame.

2.3.1. *Reference displacement between consecutive video frames*

Displacement of the trunk and average displacement of normally matched fruit were sequentially used as reference displacement to predict fruit positions between consecutive video frames. Euclidean distance between trunk positions in consecutive video frames is calculated as represented in Eq. (1).

As relative positions of trunks in different video frames do not change significantly, trunks with the minimum Euclidean distance in consecutive video frames were regarded as the same trunk and tracked to obtain reference displacement in mutual match. Tracked trunk in the initial video frame, which was first detected trunk in the direction of video motion, was subsequently transformed into another detected trunk when the currently tracked trunk reaches the left boundary of the video field of view. It can ensure the consistency of the choice of trunk sections between consecutive video frames, as shown in Fig. 5. In case of no detected trunk, average reference displacement of previous five video frames was considered as reference displacement. In secondary match, displacements of normally matched fruit between the previous frame and the current frame were averaged as the new reference displacement.

$$Ed = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (1)$$

where x_i and y_i refer to the horizontal and vertical coordinates of the centre position of detected object bounding boxes in video frame i, respectively.

2.3.2. *Mutual match*

One-way fruit match with the minimum Euclidean distance, where only predicted fruit is matched to detected fruit or detected fruit is matched to predicted fruit in the current video frame, may cause mismatch of the clustered fruit. Predicted fruit of the current video frame was obtained based on detected fruit of the previous video frame and reference displacement between the previous video frame and the current video frame. As shown in Fig. 6(a), fruit A_d and fruit A_p are the same fruit that is detected and predicted in the current

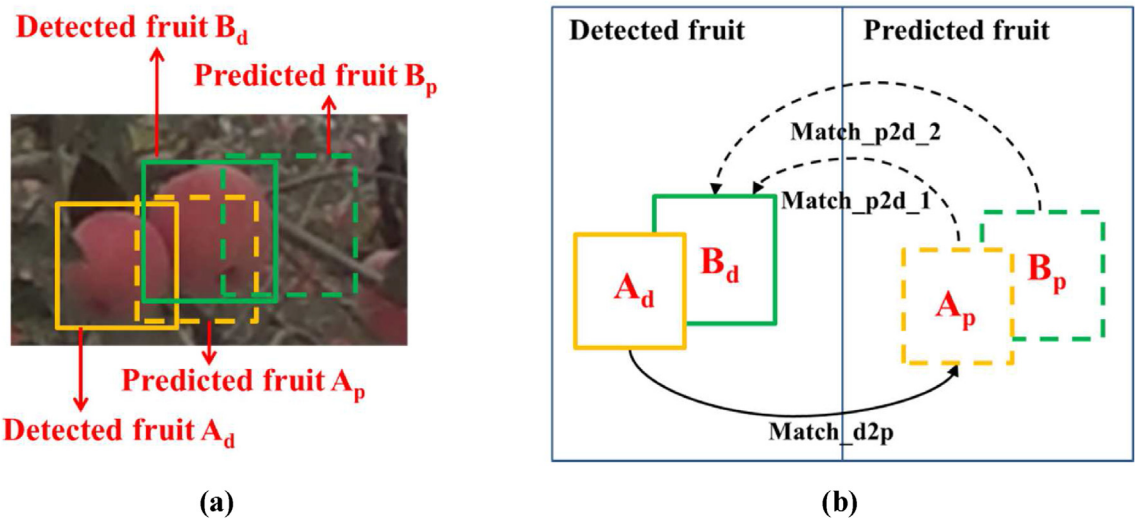


Fig. 6 – Mutual match for detected and predicted fruit. Fruit A_d, fruit A_p, fruit B_d and fruit B_p in (a) are the same as those in (b). (a) Detected and predicted fruit in the current video frame are represented by solid and dashed bounding boxes, respectively. (b) Mutual match relationships in consecutive video frames. Match_p2d_1 and Match_p2d_2 both refer to match relationships of predicted fruit to detected fruit, which are indicated by the black dashed line with arrows. Match_d2p refer to match relationship of detected fruit to predicted fruit, which is indicated by the solid black line with arrows.

video frame, respectively. Similarly, fruit B_d and fruit B_p are the same fruit that is detected and predicted in the current video frame, respectively. Fruit B_d was mismatched with fruit A_p based on one-way fruit match with the minimum Euclidean distance because it has a smaller Euclidean distance from fruit A_p than fruit B_p. Therefore, a mutual match algorithm for the predicted and detected fruit in the current video frame was proposed.

Mutual match algorithm referred to that not only predicted fruit was matched to detected fruit, but also detected fruit was matched to predicted fruit in the current video frame. Firstly, predicted fruit was matched to the detected fruit in the current video frame, and its match relationships were defined as Match_p2d_1 and Match_p2d_2, as shown in Fig. 6(b). Then detected fruit was matched to predicted fruit in the current video frame, and its match relationships were defined as Match_d2p. Finally, a match relationship competition strategy was established to obtain the priority match relationship in Match_p2d_1, Match_p2d_2 and Match_d2p.

The match relationship competition strategy was developed based on the number of match relationships. Firstly, fruit B_d of Match_p2d_2 was found, which competed with fruit A_d of Match_d2p for fruit A_p, as shown in Fig. 6(b). Then, fruit B_p of Match_p2d_1 was found, which competed with fruit A_p of Match_d2p for fruit B_d and only occurred when the clustered fruit to be matched. Finally, the Match_d2p had a competitive priority, because fruit B_d had match relationships with multiple fruit, while fruit A_d had match relationships with only one fruit. Other match relationships related to fruit A_d and fruit A_p, such as Match_p2d_1, were deleted. Fruit B_p was matched with fruit B_d based on match relationship Match_p2d_2.

2.3.3. *Abnormal match removal*

Uncertainty of orchard environment, vehicle motion, fruit detection, and trunk detection may cause abnormal match of fruit that need to be removed. Motion trajectories of all fruit should be parallel to each other in the same time period. However, many of the trajectories were not parallel due to abnormal matches. Therefore, abnormal matches, which were judged by comparing positions of mutual matched fruit between consecutive video frames, need to be removed. Theoretically, coordinate of the same fruit in the horizontal direction will keep decreasing or increasing with video motion from fruit appearance to disappearance. Due to the slow speed of the vehicle and the flat ground in this study, the coordinates of the fruit in the vertical direction will remain relatively constant. Since both width and height of a fruit in tree row videos were both about 30–40 pixels, mutual matched fruit that moved more than 30 pixels in any direction in consecutive video frames were judged as abnormal and were thus removed.

2.3.4. *Secondary match*

Both unmatched fruit and abnormally matched fruit were secondary matched based on average displacement of normally matched fruit to maximise the number of normally matched fruit after removing abnormal matches. Displacements of normally matched fruit were averaged as the new reference displacement in place of the displacement of the

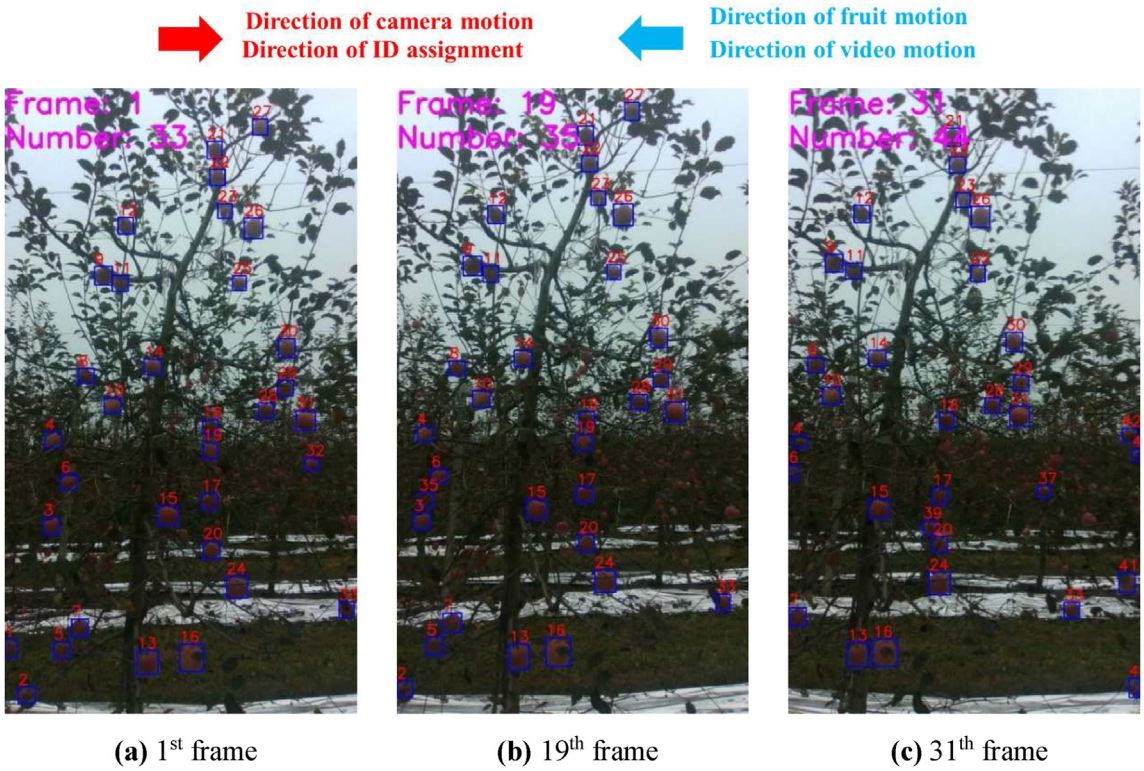


Fig. 7 – An example of fruit counting. Red arrow at the top represents directions of camera motion and ID assignment, while blue arrow represents directions of fruit and video motion; Numbers above blue bounding boxes refer to fruit IDs; The maximum number of fruit IDs in the last video frame will be fruit counting result. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

same trunk between consecutive video frames. Unmatched fruit and abnormally matched fruit in the previous video frame were predicted based on the new reference displacement and secondary matched as the same fruit in consecutive video frames if IoU is greater than 0.4, which is calculated in Eq. (2).

$$IoU = \frac{D \cap T}{D \cup T} \tag{2}$$

where D and T refer to bounding boxes of detected the predicted fruit, respectively.

2.3.5. *Fruit counting based on ID assignment*

Fruit counting was implemented by ID assignment, where each fruit in the video frames was assigned with an ID (No. 1, No. 2, No. 3, No. 4, No. 5, etc.). The maximum number of IDs in the last video frame was fruit counting result, where fruit were defined as tracked fruit in the video frames after secondary match and each of them was assigned with an ID. As shown in Fig. 7, the direction of ID assignment is the same as direction of camera motion and opposite to direction of fruit motion.

2.3.6. *Steps of twice matched fruit counting system*

Steps of the twice matched fruit counting system were as follows (assume started at video frame 1):

For video frame 1, the processing steps are ($C_{i\delta}$ refers to processing steps, i refers to video frame, δ refers to specific step):

C_{11}: Extract a RGB video frame.
C_{12}: Detect the RGB video frame by YOLOv4-tiny to obtain bounding boxes of fruit and trunk.
C_{13}: Assign fruit IDs incrementally starting from 1 according to direction of camera motion, put current detection and count results into a list L, and obtain current fruit number N_{count}.
C_{14}: Extract the bounding box of a trunk as a tracking object.
C_{15}: Draw the ID and the bounding box of the counted fruit in the video frame.

For the subsequent video frame, the processing steps are:

C_{21}: Extract an RGB video frame.
C_{22}: Detect the RGB video frame by YOLOv4-tiny to obtain rectangular boxes of fruit and trunk (Tracking of a fruit will not be maintained when this fruit is not detected for more than five consecutive frames).
C_{23}: Calculate the displacement of the same trunk between consecutive video frames as reference displacement.

C_{24}: Mutual match detected fruit and predicted fruit based on the previous video frame of fruit information in the list L and reference displacement.
C_{25}: Calculate average displacement of normally matched fruit as a new reference displacement.
C_{26}: Secondary match of unmatched fruit and abnormally matched fruit in consecutive video frames based on the new reference displacement.
C_{27}: Put the current count result N_{count} into the list L, assign the ID to tracked fruit incrementally from N_{count}+1 according to direction of camera motion.
C_{28}: Draw the ID and the bounding box of the counted fruit on the video screen.

Repeat the above steps of the subsequent video frame until the end of the video, and output the current count result N_{count}.

Twice matched fruit counting system testing was implemented on a laptop computer with NVIDIA GeForce MX250 2 GB GPU, 8 GB of RAM, and 64-bit Windows 10. Software included Python 3.8, Microsoft Visual Studio 2017, cuDNN 7.6.5, CUDA 10.0, and OpenCV 4.4.0. The twice matched fruit counting system was developed based on Python language with OpenCV library.

2.4. Evaluation indicators

Performance of the object detection model was evaluated using average precision (*AP*), mean average precision (*mAP*), and detection speed. All samples were divided into different types based on combinations of the true and predicted class, including *TP* (true positive), *FN* (false negative), and *FP* (false positive). According to the classification of samples, precision (*P*) and recall (*R*) of the object detection model are calculated in Eqs. (3) and (4), respectively. *AP* indicates global performance of the object detection model (Gao et al., 2020) and is calculated using Eq. (5) by *P* and *R*. The mean of *AP* values (*mAP*) of fruit and trunk detection, was calculated in Eq. (6).

Bounding box size of the fruit from a YOLO detector can vary between consecutive frames, which may affect the movement vector. Thus, 20 fruit in each experimental video were randomly selected for the calculation of Detection Deviation (D_{dev}). This indicator assesses the variation in bounding box size of the fruit from the emergence to the disappearance and is calculated in Eq. (7). The number of D_{dev} corresponding to each fruit was determined by the number of frames in which it appears on the video field of view, i.e., if a fruit appears in n_{fruit} frames, the number of its D_{dev} is n_{fruit}-1.

Three indicators for evaluating the performance of fruit tracking in video frames were introduced: ID Switch Rate (W_{ID}), Multiple Object Tracking Accuracy (P_{tr}), and Multiple Object Tracking Precision (P_{mt}). These indicators are calculated in Eqs. (8)–(10). Additionally, counting accuracy (P_c) is calculated in Eq. (11) to assess performance of fruit counting in videos (Gao et al., 2022).

$$P = TP/(TP + FP) \tag{3}$$

$$R = TP/(TP + FN) \tag{4}$$

$$AP = \int_0^1 P(R)\mathrm{d}R \tag{5}$$

$$mAP = \frac{1}{2}\left(AP_{fruit} + AP_{trunk}\right) \tag{6}$$

$$D_{dev} = \sqrt{(0.5wid_{i+1} - 0.5wid_i)^2 + (0.5height_{i+1} - 0.5height_i)^2} \tag{7}$$

$$W_{ID} = \frac{Q_{sh}}{S} \tag{8}$$

$$P_{tr} = \frac{M_{mat}}{S} \tag{9}$$

$$P_{mt} = \frac{M_{mat}}{T_{mat}} \tag{10}$$

$$P_c = \left(1 - \frac{|S - G_{manu}|}{G_{manu}}\right) \times 100\% \tag{11}$$

where the wid_i and $height_i$ refer to the width and height of the detected fruit bounding box in video frame i, respectively; Q_{sh} refers to the number of fruit with ID switching when the twice matched fruit counting system performed fruit counting in experimental videos. It is obtained by image counting and averaging of the fruit visible from one inter-row side by three operators and is considered valid when the variance between operators is less than 10% of the reference number of fruit with ID switching; *S* refers to the number of algorithm (twice matched fruit counting system) counted fruit in experimental videos, each of which corresponds to a different experimental video; M_{mat} and T_{mat} refer to the number of correctly tracked fruit and total number of matched fruit in experimental videos, respectively; G_{manu}, the number of ground truth in experimental videos (reference count), was obtained by the operator using DarkLabel software to mark fruit in the tree row video (fruit on the ground and in the back row of trees were not marked). The reference count (Mean ± variance) for the same tree row video will be obtained by three operators, which was considered valid when the variance between operators is less than 1% of the reference count.

3. Results and discussions

3.1. Performance of object detection model

The object detection model achieved satisfactory performance for detecting both fruit and trunks in modern apple orchards. Performance of the object detection model was tested with 320 images at confidence threshold of 0.25, which contained 17,724 'fruit' and 451 'trunk' targets of interest. The object detection model took only 16 ms on average to process an image of a resolution of 720 × 1280 pixels, indicating that object detection model quickly detected 'fruit' and 'trunk' (see Table 2). For 'fruit', it successfully detected 16,723 *TP*, with *FP* of 2684 and *FN* of 1001 (AP_{fruit} of 94.2%). For 'trunk', it

Table 2 – Detection results on the test dataset.

Object	TP	FP	FN	P/%	R/%	AP/%	mAP/%	Speed/(ms/image)
Fruit	16,723	2684	1001	86.5	94.5	94.2	96.4	16
Trunk	445	4	6			98.6		

successfully detected 445 *TP*, with *FP* of 4 and *FN* of 6 (AP_{trunk} of 98.6%). Since trunk is an extended object unlike fruit which is a discrete object, the bounding box was not fit to the whole trunk, as shown in Fig. 8(a). Despite the high mAP of 96.4%, fruit on the ground and in the back row of trees, as shown in Fig. 8(b)–(c), causes lower *P* of 86.5% (below 90%).

The lower P value could be abated by a series of threshold-limited methods to improve detection performance in future research. For example, fruit on the ground generally have low detection confidence, which could be abated by setting a confidence threshold in fruit detection (He et al., 2022; Mirhaji et al., 2021; Osman et al., 2021; Vasconez et al., 2020). Fruit in the back row of trees could also be abated by setting a distance threshold based on depth (Fu et al., 2020; Koirala et al., 2019).

The variation in the bounding box size of the fruit from a YOLO detector between consecutive frames was small, which did not significantly affect the movement vector of the fruit between consecutive frames. A total of 7658 D_{dev}s were obtained from 200 fruit in 10 experimental videos, as shown in Fig. 9. There are 5209 D_{dev}s smaller than or equal to 1 pixel, constituting over 68.0% of all D_{dev}s, indicating that the variation in the bounding box size of the fruit from a YOLO detector between consecutive frames is small in most cases. The maximum value of D_{dev}s is 5 pixels, and only 6 of them exist, which accounts for less than 0.1% of all D_{dev}s. This implies that the impact of this variation on the movement vector of the fruit between consecutive frames is very small, given that objects translate about 10–20 pixels between consecutive frames.

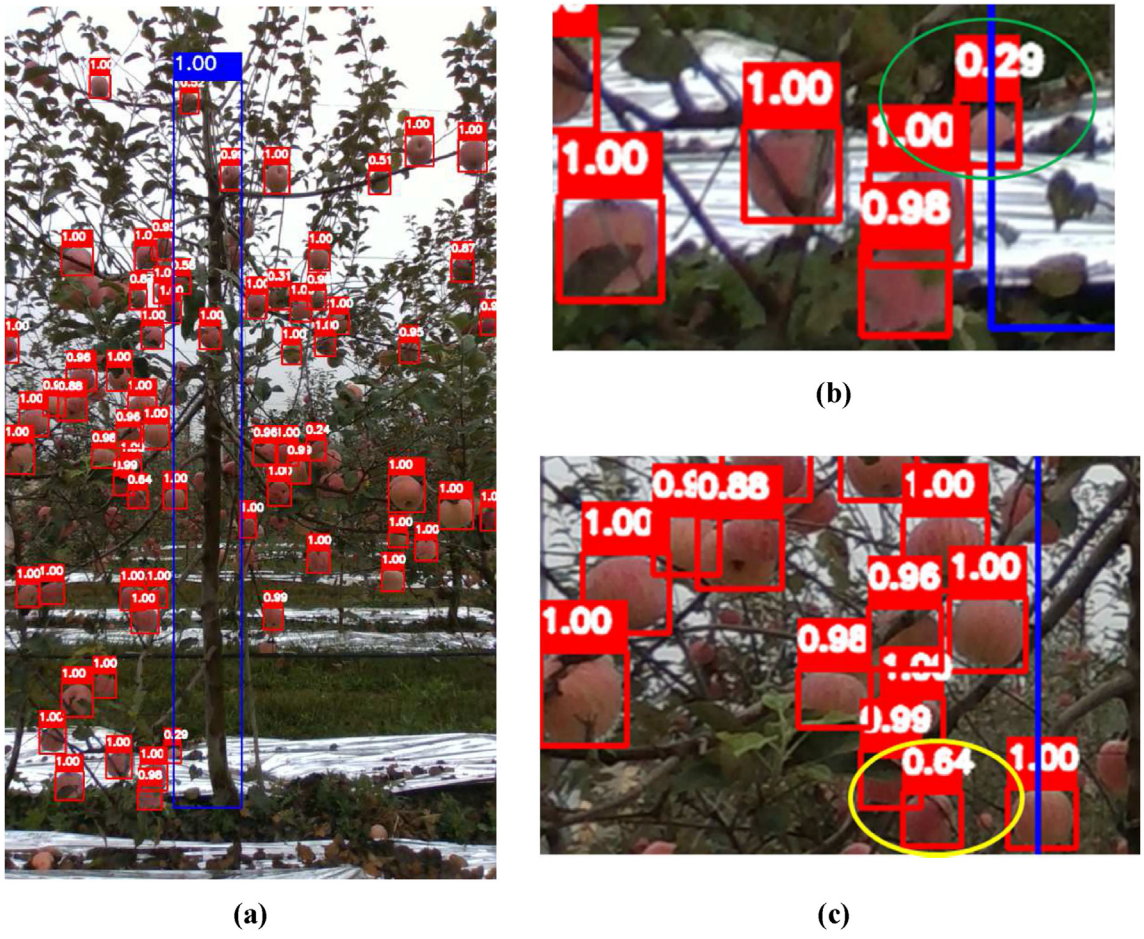


Fig. 8 – Examples of *FP* in an image of testing dataset, where the detected fruit and trunk are marked with red and blue bounding boxes, respectively. Numbers above bounding boxes represent confidence threshold of detection. (a) An example of fruit detection; (b) Fruit on the ground is in the green circle; (c) Fruit in the back row of trees is in the yellow circle. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

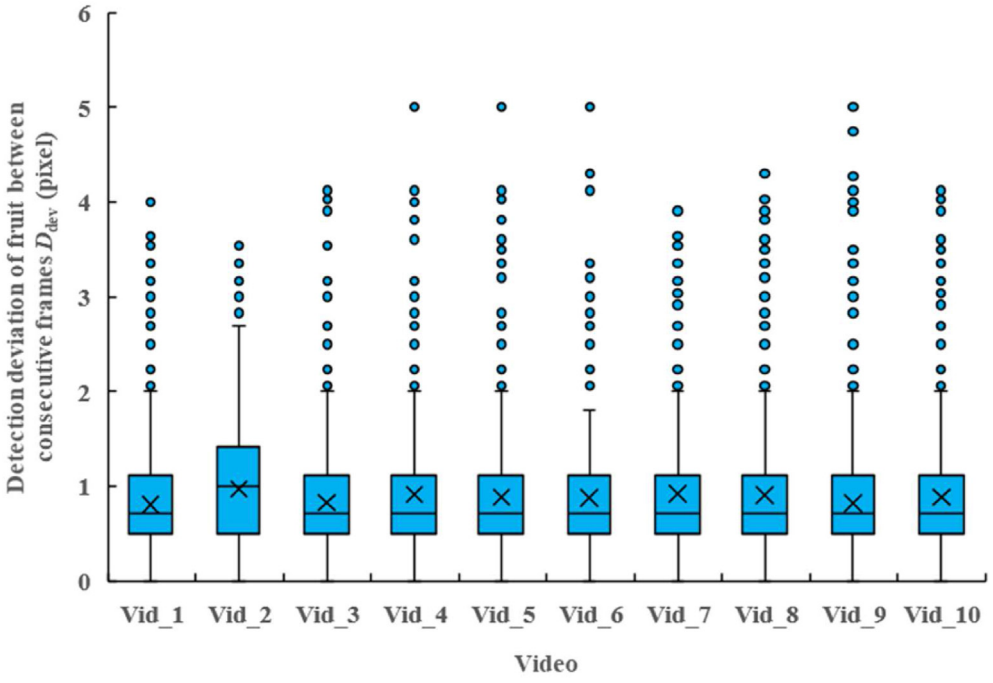


Fig. 9 – Detection deviation (D_{dev}) results of the twice matched fruit counting system for experimental videos. The circles at the top of the box plot refer the outliers.

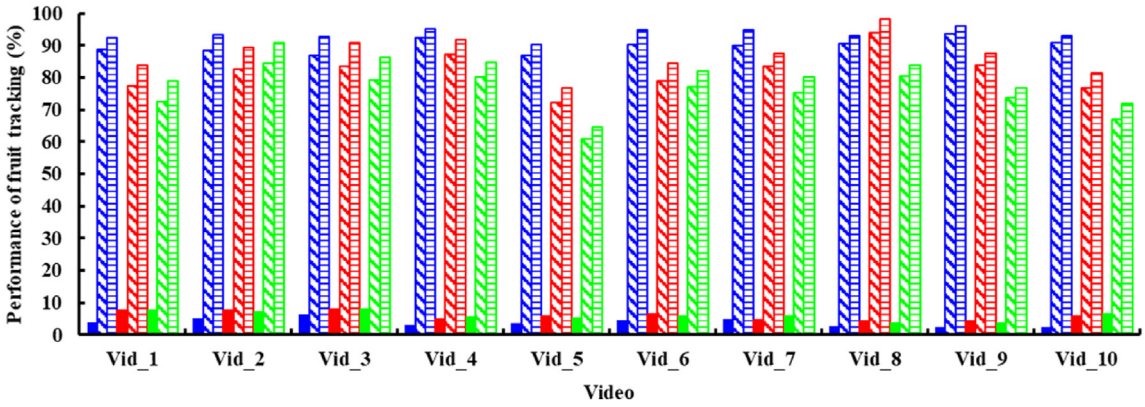


Fig. 10 – Fruit tracking results of the twice matched fruit counting system for experimental videos. Vid_1, Vid_2, Vid_3, Vid_4, Vid_5, Vid_6, Vid_7, Vid_8, Vid_9, and Vid_10 consist of 135, 295, 182, 194, 196, 176, 181, 171, 178, and 181 frames, respectively; Fruit tracking results for experimental videos with 30 fps are marked in blue; Fruit tracking results for experimental videos with 20 fps are marked in red; Fruit tracking results for experimental videos with 15 fps are marked in green; Colour filled bars refer to results of ID Switch Rate (W_{ID}); Diagonal bars refer to results of Multiple Object Tracking Accuracy (P_{tr}); Horizontal bars refer to results of Multiple Object Tracking Precision (P_{mt}). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

3.2. *Performance of fruit tracking and counting*

The twice matched fruit counting system had the potential to maintain good tracking performance for experimental videos with more separation between objects in consecutive frames. As shown in Fig. 10, the average W_{ID} associated with the twice matched fruit counting system for experimental videos with 30 fps was 3.9%. This means that most fruit (even the clustered fruit) keep their ID unchanged from appearance to disappearance. Fruit could be successfully tracked in subsequent video frames even if it was not detected in a video frame, as shown in Fig. 11 (No. 158 fruit in the 95th frame). The average P_{tr} and P_{mt} associated with the twice matched fruit counting system for experimental videos with 30 fps were 89.9% and 93.5%, respectively. The average W_{ID}, P_{tr} and P_{mt} associated with the twice matched fruit counting system for experimental videos with 20 fps were 5.9%, 82.0% and 87.1%, respectively. This means that the frame rate of the experimental video can be changed from 30 to 20 fps by dropping frames before running the twice-matched fruit counting system, which still achieved satisfactory tracking performance.

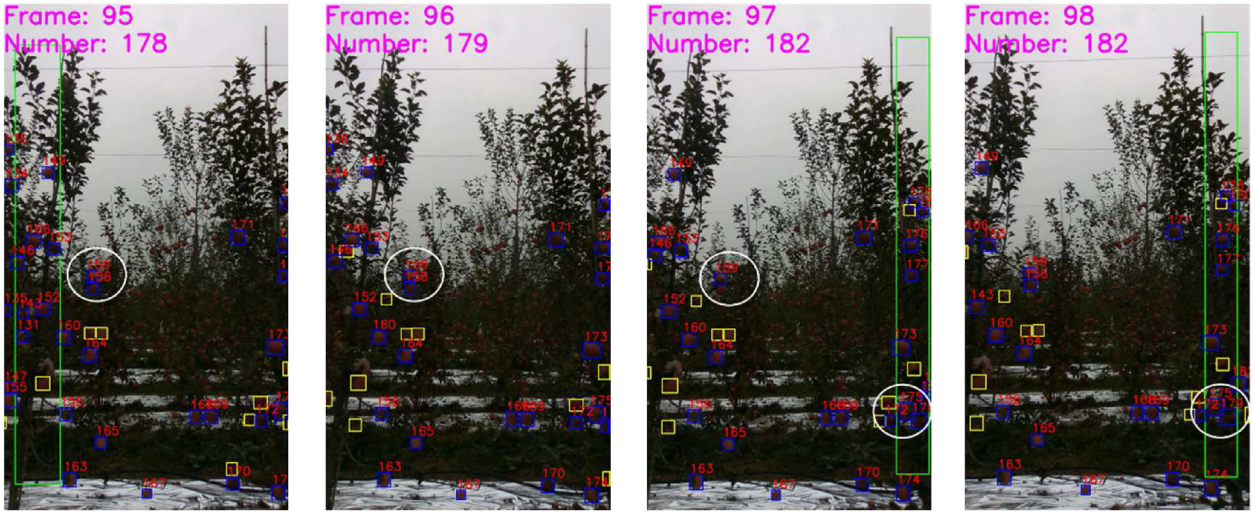


Fig. 11 – An example of the twice matched fruit counting system counts the fruit of four consecutive video frames starting from the 95th frame in the Vid_1, in which most fruit IDs remain unchanged. Tracked trunk is marked with green bounding box. Tracked and detected fruit are marked with blue and yellow bounding box, respectively. Fruit in the white circle are clustered fruit. Note: IDs of clustered fruit in the green circle are No. 158, No. 159, No. 172, No. 175, No. 179. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

Abnormal match removal ensures that the twice-matched fruit counting system maintains good tracking performance. Before removing abnormal matches, there are some significant mismatches in spatial location mainly caused by fruit that are not detected in the current frame. In mutual match, the predicted and detected fruit of the current frame will be matched based on the minimum Euclidean distance. The relative spatial positions of the fruit are invariant, which makes it possible to remove abnormal matches based on the Euclidean distance, as shown in Fig. 12(a) and (b). No. 32 fruit in 26th and 27th frames was not detected in the 28th frame due to an occlusion of a leaf and a video field of view boundary, which caused it to be mismatched to another fruit without an ID assigned, as shown in Fig. 12(a1)–(a3) and (b1)–(b3). A motion trajectory of fruit containing a significant mismatch in the spatial location before removing abnormal matches often corresponds to different fruit. After removing abnormal matches, each of these different fruit correspond to a different motion trajectory, which made 40 motion trajectories of fruit into 46.

The main reason for superior fruit tracking performance of the twice-matched fruit counting system is the consideration of orchard planting characteristics. Individual characteristics of fruit are similar and difficult to distinguish, which makes it sensible to abate mismatch caused by the clustered fruit based on match relationships in consecutive video frames.

According to the ground truth data obtained by visual manual counting, the twice matched fruit counting system showed superior counting performance compared to DeepSORT algorithm in experimental videos. Although some studies on fruit counting reported good results (Bhattarai & Karkee, 2022; Häni et al., 2020; He et al., 2022), it is difficult to evaluate proposed solutions when each solution is used with a different data set. Therefore, experimental videos were applied to compare the performance of the twice matched fruit counting system and DeepSORT algorithm. Linear regressions between algorithmic and manual count (G_{manu}) for experimental videos (30 fps) are illustrated in Fig. 13(a). Root mean squared error (*RMSE*) and the coefficient of determination (R^2) associated with the twice matched fruit counting system were 16.3 fruit/video and 0.93, respectively, indicating that the number of fruit counted was correlated to the ground truth. Whereas the *RMSE* and R^2 associated with the DeepSORT algorithm were 1242.5 fruit/video and 0.73, respectively, implying that the number of fruit counted was much more than the ground truth.

The core of the DeepSORT algorithm is the recursive Kalman filter and the Hungarian algorithm, which combines motion and appearance information to perform association of detected fruit between consecutive video frames to achieve fruit counting. However, there were no obvious differences in appearance between fruit. The motion of experimental videos became irregular due to the movement of the remote-controlled vehicle on the muddy ground. These make it difficult to combine motion and appearance information to perform association of detected fruit between consecutive video frames thus causing over-counting, as shown in Fig. 13(a1)–(a4). Counting results of DeepSORT algorithm were about 3–7 times of the ground truth, while the twice matched fruit counting system achieved an average P_c of 94.0% across all experimental videos. Similar results were found on videos of a commercial fruiting-wall 'Scifresh' apple orchard, where over-counting results were more than four times the ground truth (Gao et al., 2022). In terms of speed, the twice matched fruit counting system was implemented on CPU at 3–5 fps,

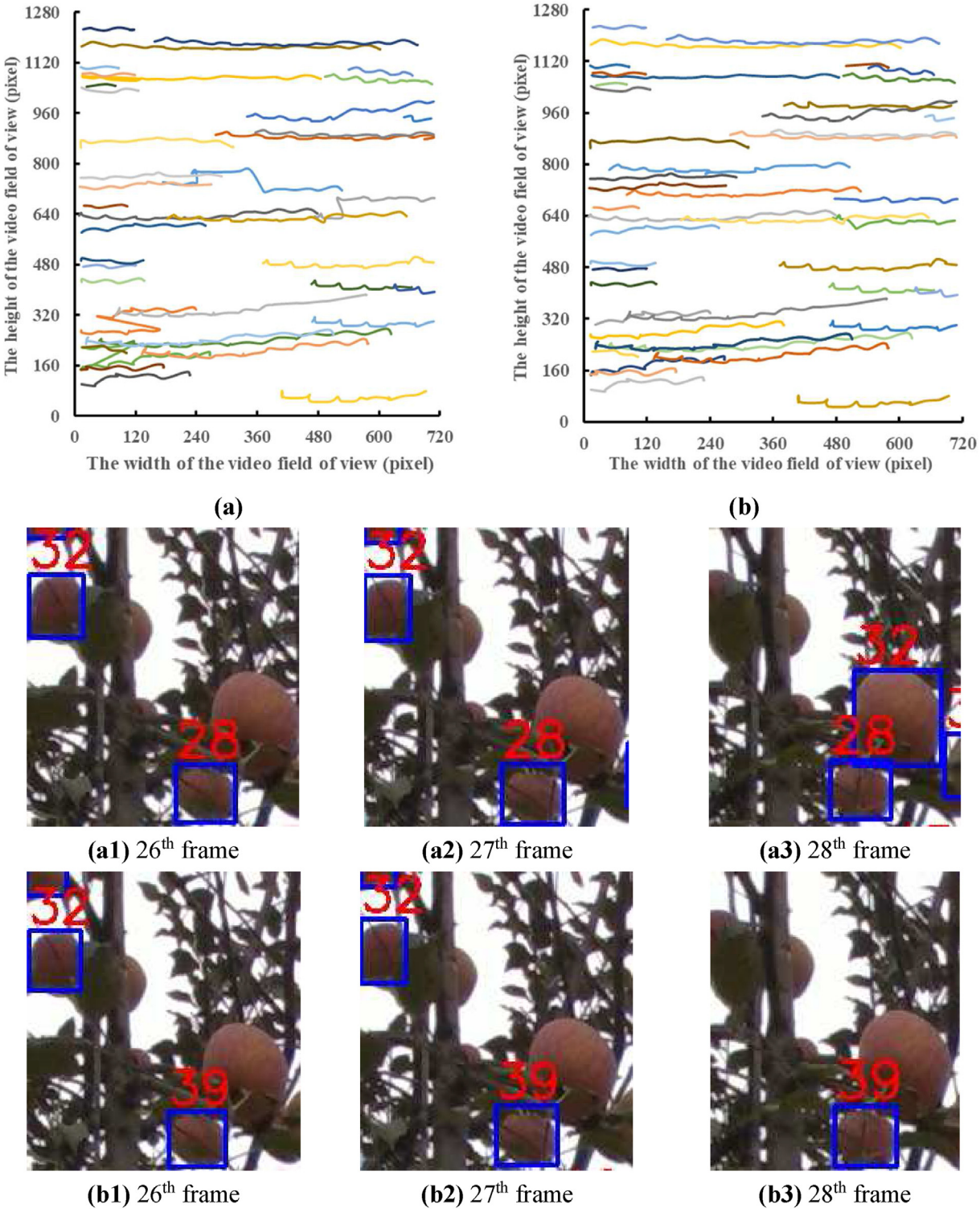


Fig. 12 – Motion trajectories of randomly selected fruit from 1st to 45th frame in the Vid_1. (a) Fruit motion trajectories before removing abnormal matches. (b) Fruit motion trajectories after removing abnormal matches. (a1), (a2), and (a3) are the abnormal match of No. 32 fruit from 26th to 28th frame, which is removed in (b1), (b2), and (b3).

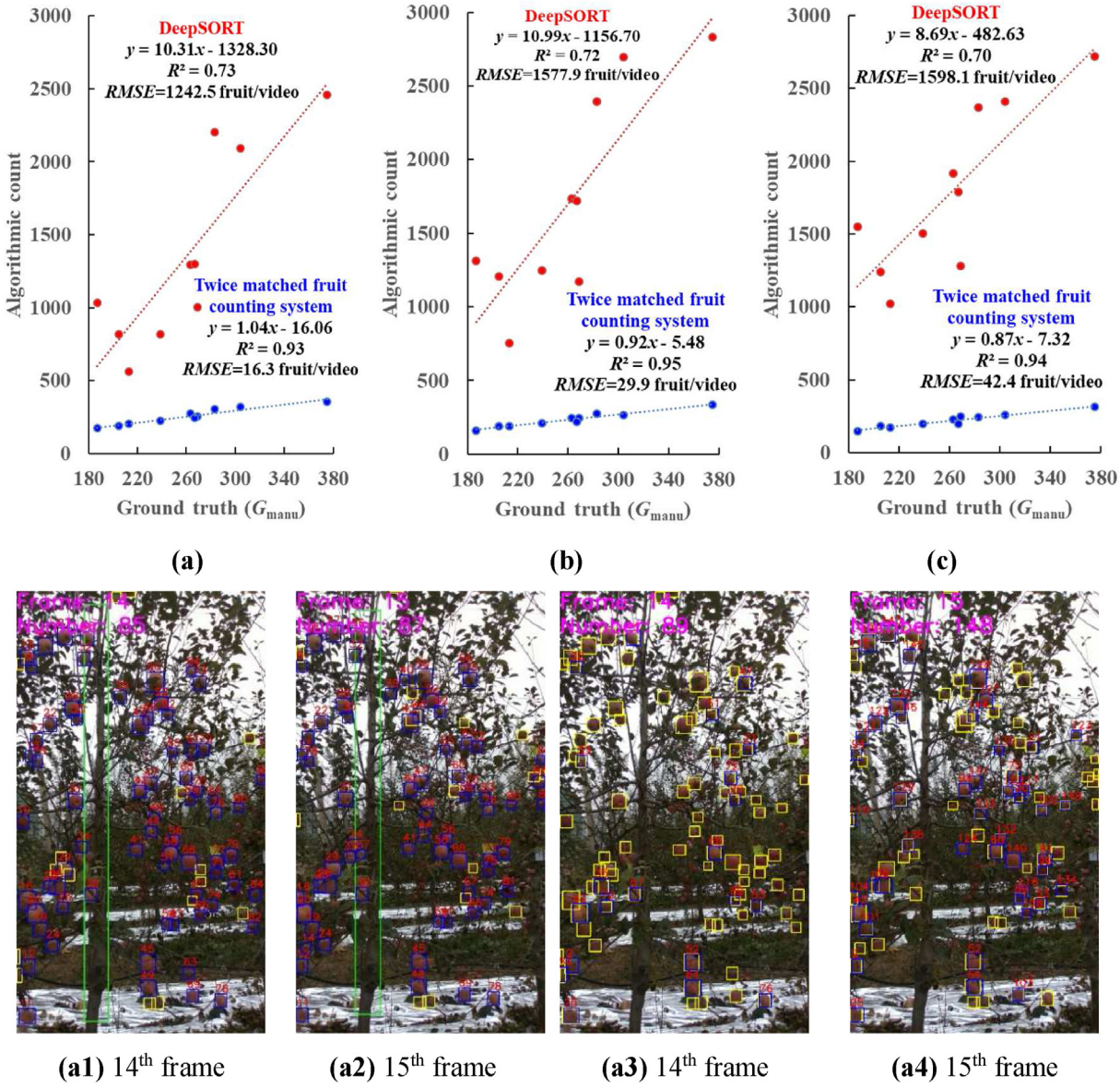


Fig. 13 – Comparison of fruit counting performance of twice matched fruit counting system and DeepSORT algorithm for experimental videos. Red and blue dots represent the counting results of the DeepSORT algorithm and the twice matched fruit counting system, respectively; The red dashed line represents a linear regression between the G_{manu} and counting results of DeepSORT algorithm; The blue dashed line represents a linear regression between the G_{manu} and counting results of the twice matched fruit counting system; G_{manu} of Vid_1, Vid_2, Vid_3, Vid_4, Vid_5, Vid_6, Vid_7, Vid_8, Vid_9, Vid_10 were 239 ± 2, 269 ± 2, 205 ± 1, 263 ± 1, 267 ± 2, 304 ± 1, 375 ± 2, 283 ± 1, 213 ± 1, 187 ± 1 respectively. (a) Linear regressions between algorithmic and manual count (G_{manu}) for experimental videos with 30 fps. (a1) and (a2) are examples of fruit counting of the twice matched fruit counting system on Vid_1; (a3) and (a3) are examples of fruit counting of DeepSORT algorithm on Vid_1. (b) Linear regressions between algorithmic and manual count (G_{manu}) for experimental videos with 20 fps. (c) Linear regressions between algorithmic and manual count (G_{manu}) for experimental videos with 15 fps. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

while DeepSORT algorithm was implemented on CPU at 1–2 fps.

The twice matched fruit counting system had the potential to maintain good counting performance for experimental videos with more separation between objects in consecutive frames. The *RMSE* and R^2 associated with the twice matched fruit counting system for experimental videos with 20 fps were 29.9 fruit/video and 0.95, respectively, as shown in Fig. 13(b). Whereas the counting performance of the DeepSORT algorithm further degraded with the *RMSE* and R^2 of 1577.9 fruit/video and 0.72, respectively. When the frame rate of experimental videos was 15 fps, the *RMSE* and R^2 associated with the twice matched fruit counting system were 42.4 fruit/video and 0.94, respectively, as shown in Fig. 13(c). The average P_c of the two-match fruit counting system were 89.4% and 84.4% for the experimental videos with frame rates of 20

and 15 fps, respectively. This means that the frame rate of the experimental video can be changed from 30 to 20 fps by dropping frames before running the twice-matched fruit counting system, which still achieved satisfactory counting performance.

Video-based counting method, in which image sequences are collected from multiple viewpoints to observe tree row (Gao et al., 2022; Zhang et al., 2022), shows promise for the twice matched fruit counting system to be applied to actual orchards. The twice matched fruit counting system showed superior counting performance in experimental videos, however, its generalisation capabilities need to be verified with diverse dataset.

4. Conclusions

In this study, the twice matched fruit counting system was proposed for apple fruit counting, which includes the object detection model based on YOLOv4-tiny, fruit tracking with mutual match, and fruit counting with ID assignment. Promising results for fruit tracking and counting were obtained, illustrating the superiority of mutual match algorithm in video-based fruit counting. Most fruit keep their ID unchanged from the appearance to the disappearance, indicating that mutual match algorithm alleviated match errors associated with the clustered fruit. Abnormal matches were also removed after mutual matching, which helped to achieve smooth trajectories.

Although, the twice matched fruit counting system showed superior performance relative to visual manual counts, performance of the object detection model impacted because of the high *FP* (false positive, including fruit on the ground or in the back row of trees). These *FP* could be abated to improve detection performance in future research. This study only focuses on the mature fruit, whereas, fruit counting is also desirable to include immature fruit at fruitlet stage. Hence, new algorithms for fruit counts in other growth stages could be developed in the future. For immature fruit at the mature stage, it is possible to distinguish them from mature fruit if hyperspectral data that can reflect the physiological characteristics of the fruit is collected. The assumption that 'the coordinates of the fruit in the vertical direction will remain relatively constant' in this study is an artefact of the current experimental condition, i.e., slow speed and flat ground. In practice, both vertical image displacements due to higher speed and uneven ground and the effect of vehicle turning relative to the row on apparent fruit trajectory can make fruit counting difficult. Therefore, in the future, we will perform fruit counting based on instance segmentation and 3D reconstruction techniques by projecting the fruit into a 3D reconstruction model of the tree rows.

With continuous work, the twice matched fruit counting system has great potential to support farmers with a smart and precise surveillance approach of fruit at different growth stages, and therefore leads to better orchard management decisions.

Declaration of competing interest

The authors declared that we have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Acknowledgements

This work was 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 (TGZX2021-29); National Foreign Expert Project, Ministry of Science and Technology, China (DL2022172003L, QN2022172006L).

REFERENCES

Al-Sa'd, M., Kiranyaz, S., Ahmad, I., Sundell, C., Vakkuri, M., & Gabbouj, M. (2022). A social distance estimation and crowd monitoring system for surveillance cameras. *Sensors, 22*, 418. https://doi.org/10.3390/s22020418

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. *European Journal of Agronomy, 115*, Article 126030. https://doi.org/10.1016/j.eja.2020.126030

Badue, C., Guidolini, R., Carneiro, R. V., Azevedo, P., Cardoso, V. B., Forechi, A., Jesus, L., Berriel, R., Paixão, T. M., Mutz, F., de Paula Veronese, L., Oliveira-Santos, T., & De Souza, A. F. (2021). Self-driving cars: A survey. *Expert Systems with Applications, 165*, Article 113816. https://doi.org/10.1016/j.eswa.2020.113816

Behera, S. K., Rath, A. K., & Sethy, P. K. (2021). Fruits yield estimation using Faster R-CNN with MIoU. *Multimedia Tools and Applications, 80*(12), 19043–19056. https://doi.org/10.1007/s11042-021-10704-7

Bhattarai, U., & Karkee, M. (2022). A weakly-supervised approach for flower/fruit counting in apple orchards. *Computers in Industry, 138*, Article 103635. https://doi.org/10.1016/j.compind.2022.103635

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 Robotics and Automation Letters, 2*(2), 781–788. https://doi.org/10.1109/LRA.2017.2651944

Dorj, U. O., Lee, M., & Yun, S. seok (2017). An yield estimation in citrus orchards via fruit detection and counting using image processing. *Computers and Electronics in Agriculture, 140*, 103–112. https://doi.org/10.1016/j.compag.2017.05.019

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

Gao, F., Fang, W., Sun, X., Wu, Z., Zhao, G., Li, G., Li, R., Fu, L., & Zhang, Q. (2022). A novel apple fruit detection and counting methodology based on deep learning and trunk tracking in modern orchard. *Computers and Electronics in Agriculture, 197*, Article 107000. https://doi.org/10.1016/j.compag.2022.107000

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. *Computers and Electronics in Agriculture, 176*, Article 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. *Transactions of the Chinese Society of Agricultural Engineering, 37*(21), 217–224.

Häni, N., Roy, P., & Isler, V. (2020). A comparative study of fruit detection and counting methods for yield mapping in apple orchards. *Journal of Field Robotics, 37*(2), 263–282. https://doi.org/10.1002/rob.21902

He, L., Wu, F., Du, X., & Zhang, G. (2022). Cascade-SORT: A robust fruit counting approach using multiple features cascade matching. *Computers and Electronics in Agriculture, 200*, Article 107223. https://doi.org/10.1016/j.compag.2022.107223

Jiang, C., Ren, H., Ye, X., Zhu, J., Zeng, H., Nan, Y., Sun, M., Ren, X., & Huo, H. (2022). Object detection from UAV thermal infrared images and videos using YOLO models. *International Journal of Applied Earth Observation and Geoinformation, 112*, Article 102912. https://doi.org/10.1016/j.jag.2022.102912

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

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. *European Conference on Computer Vision*, 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

Liu, X., Chen, S. W., Liu, C., Shivakumar, S. S., Das, J., Taylor, C. J., Underwood, J., & Kumar, V. (2019). Monocular camera based fruit counting and mapping with semantic data association. *IEEE Robotics and Automation Letters, 4*(3), 2296–2303. https://doi.org/10.1109/LRA.2019.2901987

Lukežič, A., Vojíř, T., Čehovin Zajc, L., Matas, J., & Kristan, M. (2018). Discriminative correlation filter tracker with channel and spatial reliability. *International Journal of Computer Vision, 126*(7), 671–688. https://doi.org/10.1007/s11263-017-1061-3

Maldonado, W., & Barbosa, J. C. (2016). Automatic green fruit counting in orange trees using digital images. *Computers and Electronics in Agriculture, 127*, 572–581. https://doi.org/10.1016/j.compag.2016.07.023

Massah, J., Asefpour Vakilian, K., Shabanian, M., & Shariatmadari, S. M. (2021). Design, development, and performance evaluation of a robot for yield estimation of kiwifruit. *Computers and Electronics in Agriculture, 185*, Article 106132. https://doi.org/10.1016/j.compag.2021.106132

Ma, Y., Zhang, Z., Kang, Y., & Özdoğan, M. (2021). Corn yield prediction and uncertainty analysis based on remotely sensed variables using a Bayesian neural network approach. *Remote Sensing of Environment, 259*, Article 112408. https://doi.org/10.1016/j.rse.2021.112408

Mekhalfi, M. L., Nicolò, C., Ianniello, I., Calamita, F., Goller, R., Barazzuol, M., & Melgani, F. (2020). Vision system for automatic on-tree kiwifruit counting and yield estimation. *Sensors, 20*, 4214. https://doi.org/10.3390/s20154214

Meng, X., Li, Y., Yuan, Y., Zhang, Y., Li, H., Zhao, J., & Liu, M. (2020). The regulatory pathways of distinct flowering characteristics in Chinese jujube. *Horticulture Research, 7*, 123. https://doi.org/10.1038/s41438-020-00344-7

Mirhaji, H., Soleymani, M., Asakereh, A., & Abdanan Mehdizadeh, S. (2021). Fruit detection and load estimation of an orange orchard using the YOLO models through simple approaches in different imaging and illumination conditions. *Computers and Electronics in Agriculture, 191*, Article 106533. https://doi.org/10.1016/j.compag.2021.106533

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

Osman, Y., Dennis, R., & Elgazzar, K. (2021). Yield estimation and visualization solution for precision agriculture. *Sensors, 21*(19), 6657. https://doi.org/10.3390/s21196657

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. *Precision Agriculture, 18*(2), 224–244. https://doi.org/10.1007/s11119-016-9458-5

Rahnemoonfar, M., & Sheppard, C. (2017). Deep count: Fruit counting based on deep simulated learning. *Sensors, 17*, 905. https://doi.org/10.3390/s17040905

Roy, P., Kislay, A., Plonski, P. A., Luby, J., & Isler, V. (2019). Vision-based preharvest yield mapping for apple orchards. *Computers and Electronics in Agriculture, 164*, Article 104897. https://doi.org/10.1016/j.compag.2019.104897

Sheng, Y., Hao, Z., Peng, Y., Liu, S., Hu, L., Shen, Y., Shi, J., & Chen, J. (2021). Morphological, phenological, and transcriptional analyses provide insight into the diverse flowering traits of a mutant of the relic woody plant Liriodendron chinense. *Horticulture Research, 8*, 174. https://doi.org/10.1038/s41438-021-00610-2

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

Vasconez, J. P., Delpiano, J., Vougioukas, S., & Auat Cheein, F. (2020). Comparison of convolutional neural networks in fruit detection and counting: A comprehensive evaluation. *Computers and Electronics in Agriculture, 173*, Article 105348. https://doi.org/10.1016/j.compag.2020.105348

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. *Precision Agriculture, 19*(6), 1062–1083. https://doi.org/10.1007/s11119-018-9574-5

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

Wang, C., Wang, X. C., Wang, Z., Zhu, W. Q., & Hu, R. (2022). COVID-19 contact tracking by group activity trajectory recovery over camera networks. *Pattern Recognition, 132*, Article 108908. https://doi.org/10.1016/j.patcog.2022.108908

Yang, B., & Xu, Y. (2021). Applications of deep-learning approaches in horticultural research: A review. *Horticulture Research, 8*, 123. https://doi.org/10.1038/s41438-021-00560-9

Zhang, Y., Peng, J., Yuan, X., Zhang, L., Zhu, D., Hong, P., Wang, J., Liu, Q., & Liu, W. (2021). MFCIS: An automatic leaf-based identification pipeline for plant cultivars using deep learning and persistent homology. *Horticulture Research, 8*, 172. https://doi.org/10.1038/s41438-021-00608-w

Zhang, W., Wang, J., Liu, Y., Chen, K., Li, H., Duan, Y., Wu, W., Shi, Y., & Guo, W. (2022). Deep-learning-based in-field citrus fruit detection and tracking. *Horticulture Research, 9*, Article uhac003. https://doi.org/10.1093/hr/uhac003