Computers and Electronics in Agriculture 237 (2025) 110476

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Original papers

Row-based kiwifruit counting pipeline for smartphone-captured videos using fruit tracking and detection region adaptation guided by support-post

Jiwei Zhang [a], Liguo Jiang [a], Leilei He [a], Zhenchao Wu [a,e], Rui Li [a,*], Jinyong Chen [f,g,h], Xiaoxu Sun [f,g,h], Yunfei Xue [i], Anastasia Grecheneva [j], Spyros Fountas [k], Longsheng Fu [a,b,c,d,*]

[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] Northwest A&F University Shenzhen Research Institute, Shenzhen, Guangdong 518000, China
[e] School of Engineering, Anhui Agricultural University, Hefei, Anhui 230061, China
[f] Zhengzhou Fruit Research Institute, Chinese Academy of Agricultural Sciences, Zhengzhou, Henan 450009, China
[g] National Key Laboratory for Germplasm Innovation & Utilization of Horticultural Crops, Zhengzhou, Henan 450009, China
[h] Zhongyuan Research Center, Chinese Academy of Agricultural Sciences, Xinxiang, Henan 453500, China
[i] Meixian Agricultural Machinery Technology Extension Service Center, Baoji, Shaanxi 722300, China
[j] Russian State Agrarian University - Moscow Timiryazev Agricultural Academy, Moscow 127434, Russia
[k] Agricultural University of Athens, Athens 11855, Greece

ARTICLE INFO

Keywords:
Yield estimation
Fruit detection
Regional detection
Multi-object tracking
Fruit counting

ABSTRACT

Automated kiwifruit counting in orchards delivers accurate, timely, and cost-effective insights into yield estimation, which is crucial for decision-making in harvesting, storage, and marketing operations. Although several studies have proposed methods for kiwifruit counting in orchard, these methods have mostly focused on a small section within the orchard row, which may provide insufficient data for practical application due to the complex and uneven fruit-growing condition. This study introduces a novel automatic approach for row-based accurate kiwifruit counting on video sequences. The sequences are captured by smartphones mounted on a stabilizer and an extension pole, from an upward perspective that spans the full length of each kiwifruit row, addressing the limitations of previous methods. The pipeline comprises of three key components: kiwifruit and support-post detection, video-based kiwifruit counting, and detection region adaptation. First, the performance of You Only Look Once (YOLO) detection models was compared based on the constructed dataset, indicating that the medium-scale model achieved a balanced performance in terms of parameters, model size, inference time and average precision (*AP*). Second, a two-containers verification (TCV) method was proposed and applied following fruit tracking to reduce over-estimation in video-based counting. Finally, the kiwifruits in neighboring rows were eliminated by detection region adaptation, which estimated the row boundaries and dynamically adapting the masks based on support-posts in orchards. YOLOv5m and YOLOv8m were considered as the most competitive frameworks in kiwifruit detection, achieving $AP_{0.5:0.95}$ scores of 0.864 and 0.878, respectively. The proposed TCV method suppressed false counts and counting accuracy improved by 59.06 % (from 40.12 % to 94.20 %) on ByteTrack and 54.08 % (from 37.59 % to 96.65 %) on DeepSORT. Moreover, the detection region adaptation guided by support-post has eliminated most fruit counts in neighboring rows. The *R*-squared (R^2) of the row-based kiwifruit counting was 0.9791, indicating the proposed approach has the potential to achieve yield estimation for kiwifruit orchards.

1. Introduction

Accurate fruit counting in kiwifruit orchards is essential for yield estimation, which provides support for both growers and agricultural cooperatives to make better decisions about resource allocation such as labor, transportation and storage. Kiwifruits are planted in rows and are trellised to grow along wires held up by support-posts and cantilever beams (Williams et al., 2019), which forms operating space for orchard

* Corresponding authors.
E-mail addresses: ruili1216@nwafu.edu.cn (R. Li), fulsh@nwafu.edu.cn (L. Fu).

https://doi.org/10.1016/j.compag.2025.110476
Received 18 January 2025; Received in revised form 31 March 2025; Accepted 23 April 2025
Available online 15 May 2025
0168-1699/© 2025 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.



Fig. 1. Tightly planted rows in the kiwifruit orchard.

management such as yield estimation. In 2023, China remains the highest yield of approximately 2.36 × 10^6 tons from a harvested area of 1.99 × 105 ha (Food and Agriculture Organization of the United Nations, 2024). Accurate automated methods can provide better guidance than rough empirical estimation and time-consuming manual counting for harvest schedules and cultivation practices (Rahnemoonfar and Sheppard, 2017).

In recent years, there has been exploratory advancement in automated kiwifruit counting based on deep learning (He et al., 2022a; Massah et al., 2021; Zhou et al., 2020). However, these methods have not been widely adopted due to challenges such as the inability to achieve complete statistical sampling, the expense of equipment, or highly standardized orchard requirements. Therefore, acquiring a complete and accessible row-based kiwifruit count for growers remains a challenge. With increasing yield and decreasing labor availability, there is a growing demand for low-cost, precise orchard management and automation (Tauber et al., 2023). Smartphones have been widely accessible in rural areas at no additional cost. The advanced sensors equipped in smartphones have the potential to become powerful tools for orchard management and data acquisition. Their cameras possess the ability to record videos in orchards, facilitating the acquisition of precise information. In this context, there is a need to develop an automated row-based counting pipeline to handle the videos captured from kiwifruit orchards.

Kiwifruit detection extracts fruit features from images, performing accurate identification and localization. Traditional image processing approaches have typically been employed to segment fruits from the background based on features such as color, shape and texture (Fu et al., 2018). However, these methods have limited robustness due to the challenges associated with feature extraction in complex and variable orchard environments. With the development of deep learning, deep convolutional neural networks (DCNN) have demonstrated high accuracy, making them a popular choice for fruit detection applications. Zhou et al. (2020) trained different models to detect kiwifruits with the objective of acquiring fruit counts, resulting in a substantial improvement in accuracy of almost 91 % when compared to traditional approaches. Although popular You Only Look Once (YOLO) models (Gao et al., 2022) facilitate the inference of images, fruit counts in images cannot provide comprehensive and detailed data for orchards due to varying fruit densities in different row sections. Thus, it can be posited that fruit counting using video-based methods is a more reasonable approach than image-based methods.

Video-based fruit counting has typically been performed through tracking-by-detection approaches, which establish correlations between instances across video sequences. The algorithms based on Kalman filter have been widely used for fruit tracking. Guo et al. (2023) employed YOLOv7 and DeepSORT to implement video-based kiwifruit counting, with video samples captured at a close distance within a small section of the orchard. Wang et al. (2019) estimated the mango fruit load using Kalman filter and Hungarian algorithm based on MangoYOLO (Walsh and Mccarthy, 2019) detections, resulting in 9.7 % repeat count and 7.1 % miss count in 110 video frames. He et al. (2022b) developed a robust fruit counting approach combining YOLOv3 detections with Kalman filter and cascade matching, leading to positive errors of 15.79 % and 6.16 % in fruit counting of camellia and apple, respectively. Parico and Ahamed (2021) performed pear counting based on YOLOv4 and DeepSORT with a false negative rate of 11.32 % and a false positive rate of 13.21 %. These methods have been found to be highly impacted by environmental conditions, which produced inaccurate over-estimation due to the inevitable accumulation of ID assignment errors.

Due to the relative motion between the object and the camera in both traffic counting and fruit counting, virtual line crossing-based methods from traffic counting have the potential to improve the fruit counting accuracy. Xu et al. (2017) developed an algorithm that determines the number and direction of vehicles whose centers hit two preset detection lines sequentially, resulting in a true detection rate ranging from 91.6 % to 95.2 %. Liang et al. (2020) proposed a vehicle counting system to determine driving directions by setting detection lines with the true detection rate maintaining approximately 90 %. Lin et al. (2022) set virtual detection regions for road sections, achieving a true detection rate of 99 % for real-time vehicle counting at lane level. These methods suggest that excess counts caused by the accumulation of the false positives can be filtered out by crossing two detection lines. This indicates that actual kiwifruits will remain tracked and counted when they cross similar preset lines or regions. It should be noted that the counting lines were predefined manually, which means that the attributed rows of the kiwifruits will not be determined automatically. Due to the tightly planting rows in the kiwifruit orchard illustrated in Fig. 1, the images and videos captured by smartphones often include the fruits in the neighboring rows. These fruits are also tracked and counted, resulting in a misattributed count.

Identifying row boundary for counting helps reduce misattributed counts, which makes the results focus on its interest. Some researchers attempted to apply DCNN model for road segmentation (Das et al., 2021;

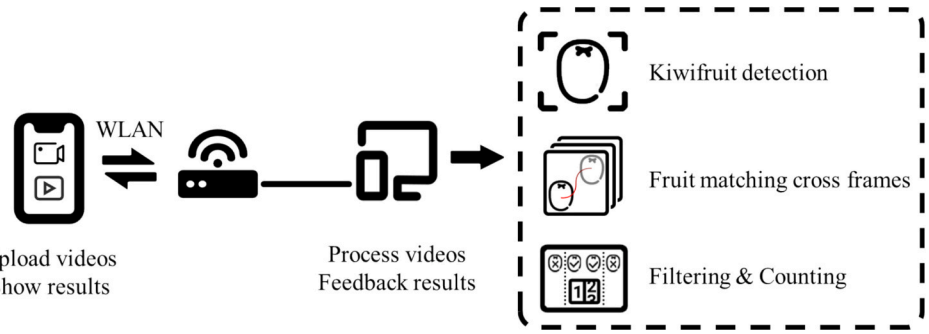


Fig. 2. The workflow of row-based kiwifruit counting pipeline.

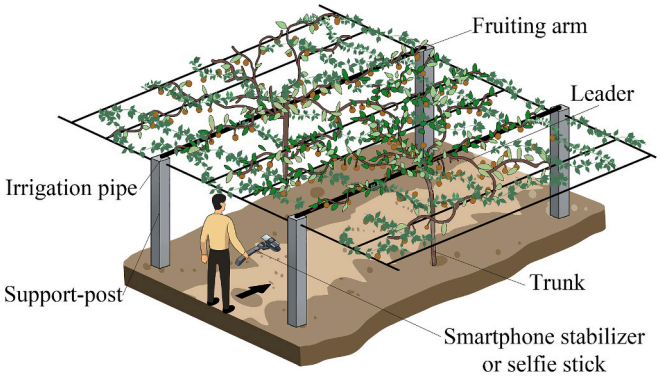


Fig. 3. Schematic of videos acquisition using smartphone in kiwifruit orchard.

Li et al., 2022), which provides explicit road boundaries based on continuous features. Li et al. (2015) proposed an extension of open active contour model with parallel constraints to handle lanes with discontinuous boundaries, achieving a true detection rate of 98 %. Zhang et al. (2018) identified the crop row with a 0.5° angular bias using plants segmentation and position clustering. In kiwifruit orchards, concrete support-posts are readily detectable and have the potential to serve as reference points for determining the row boundaries, providing the basis for detection region adaptation. Therefore, effective row boundary estimation has the potential to eliminate fruits from neighboring rows, enabling accurate row-based kiwifruit counting.

This study proposed an automatic fruit counting pipeline for kiwifruit orchards, which achieved the row-based counting of kiwifruit and eliminated abnormal fruit counts and designed a client / server (C/S) application to deploy on smartphones through wireless local area network (WLAN). The pipeline detected and tracked kiwifruit captured by videos. Then, a counting verification method was developed to reduce the over-estimation from unstable tracking. At the end, the fruits from neighboring rows were eliminated according to the estimated row-

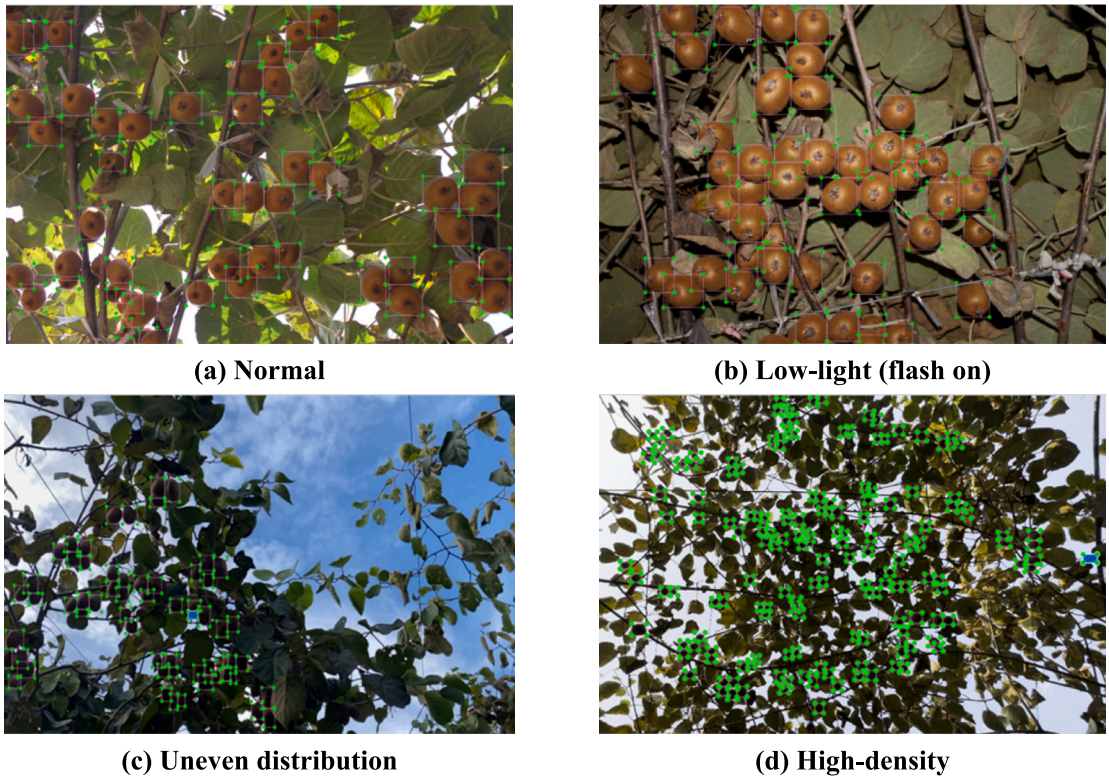


Fig. 4. Examples of labeled dataset images. The kiwifruits were marked by rectangles using LabelImg, with their vertices highlighted in green. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

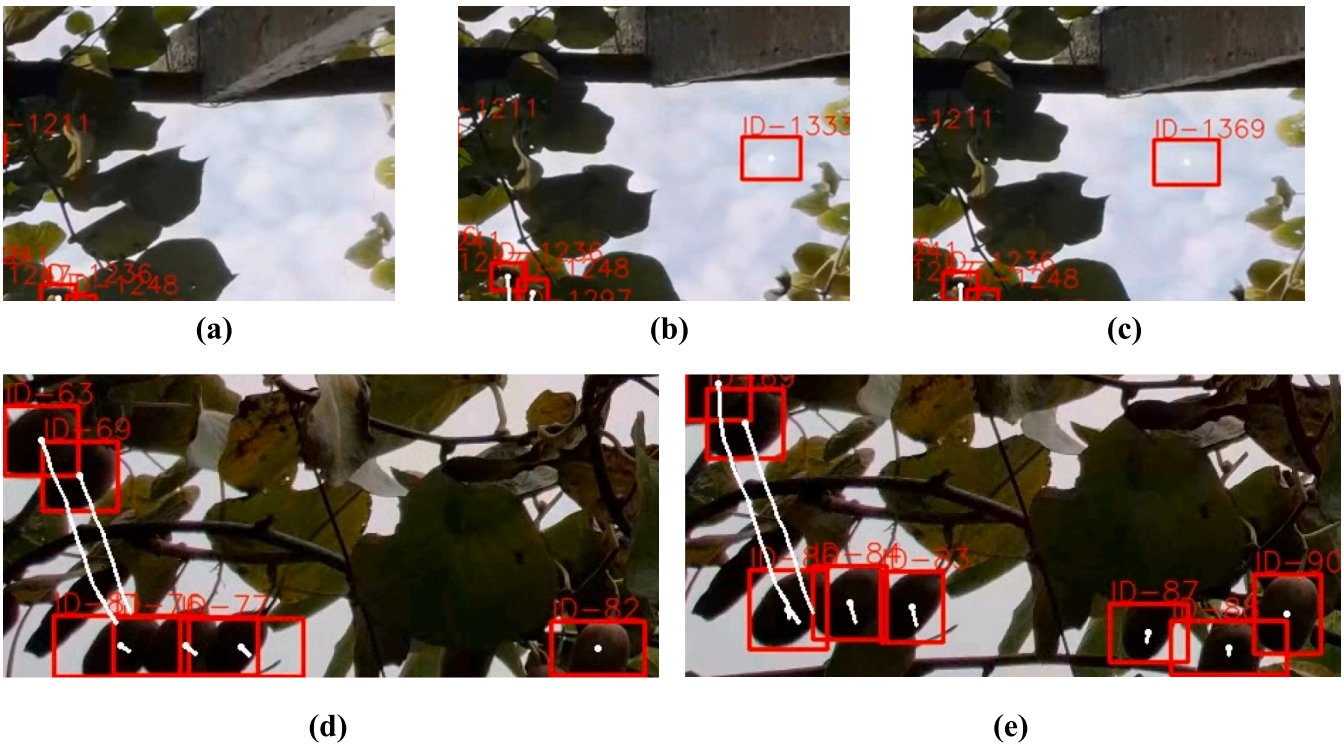


Fig. 5. The tracker registers sporadic FPs in detectors as normal kiwifruits, and the counts become skewed as the errors accumulate. (a) Normal tracking state with accurate detection in the i^{th} frame of a video; (b) an example where a FP was incorrectly assigned a track-ID in the $(i + 7)^{th}$ frame; (c) another instance of a FP being assigned a wrong track-ID in the $(i + 11)^{th}$ frame; (d) the fruits at the bottom have been assigned track-IDs in the j^{th} frame of another video; (e) the fruits were over-assigned different track-IDs in the $(j + 5)^{th}$ frame.

boundaries.

2. Materials and methods

The study aims to develop an automatic pipeline for row-based kiwifruit counting in structured orchards. Firstly, the dataset for kiwifruit detection and counting was collected and built. Secondly, different YOLO models were trained and compared to detect both fruits and support-posts, enabling accurate fruit counting and row boundary estimation. Thirdly, ByteTrack and DeepSORT were employed to match kiwifruits across video frames based on YOLOv5m detections, and a two-container verification (TCV) method was proposed to reduce over-estimation in video-based counting. Finally, the row-based kiwifruit counting pipeline integrating fruit tracking with detection region adaptation guided by support-post was implemented. The application based on C/S architecture was developed, which constituted the pipeline shown in Fig. 2. The details and metrics are presented below.

2.1. Dataset for detection

A 3-axis smartphone stabilizer, AOCHUAN Smart XE (Zhongshan Yang Guo Electronic Technology Co., Ltd., Zhongshan, PRC), along with a 73 cm extension pole, was used to stabilize the smartphone during the video collection, as shown in Fig. 3. The wide-angle camera of the smartphone captured the entire span of each kiwifruit row. Videos were recorded using a OnePlus 7 Pro (OnePlus Inc., Shenzhen, PRC), an iPhone 11 and an iPhone 14 Pro Max (Apple Inc., California, USA) at a resolution of 1920 × 1080 with a frame rate of 60 FPS. Images with a resolution of 2352 × 1568 were captured by a Canon EOS Kiss X3 (Canon Inc., Tokyo, JP), while additional images were extracted from videos in various orchards. The height of kiwifruit clusters was approximately 1.7 m, with a row spacing of 3.0 m. The kiwifruits naturally hang down in clusters when ripe, were captured by camera under canopy (Zhou et al., 2020).

A total of 1,353 images were collected and annotated, and 11 video sequences were reserved for the kiwifruit counting test. The images shot at close proximity to the canopy, as shown in Fig. 4a, provide detailed texture information for kiwifruits and background. Under conditions of moderately inadequate lighting, a flashlight ensured the visibility of the kiwifruit, as depicted in Fig. 4b. Each kiwifruit occupies more than 1.0 % of the total pixels in Fig. 4a and Fig. 4b. However, images captured by smartphones in the orchards with wide row spacing lack clear texture details. Kiwifruits are unevenly distributed within the orchard rows, as illustrated in Fig. 4c, which makes the fruit easier to be confused with leaves. This is also a typical scene with high-density of kiwifruits in smallholder orchards, as can be seen in Fig. 4d. Each kiwifruit in Fig. 4c and Fig. 4d only occupies less than 0.2 % of the image pixels. All images were randomly divided into training, validation and test sets in an 8:1:1 ratio for training the detection models. An automated labeling program was employed to improve the labeling efficiency, and LabelImg was used for manual labeling and verification. The annotations were initially saved in XML files and then converted into TXT files for model training. The dataset is available at https://github.com/fu3lab/kiwifruit_row_counting.

2.2. Fruit and support-post detection based on YOLO

Kiwifruit detection and support-post detection provided a foundation for estimating row boundaries and counting fruits. Kiwifruit detection presents unique challenges due to its classification as small-scale and intensive object detection. YOLO frames object detection as a regression problem, enabling the prediction of bounding boxes and class probabilities in a single inference, which makes it process images at high speeds. The model structures of YOLO are summarized into backbone, neck and head, with different models varying in their implementations. Key evaluation metrics, such as average precision (*AP*),

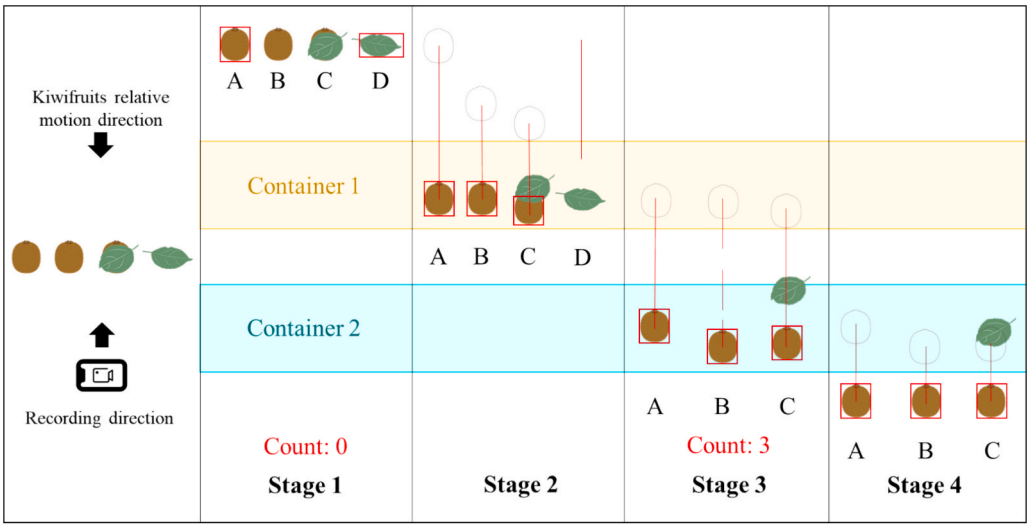


Fig. 6. Illustration for 4 different stages during counting in two-container verification (TCV) method. Case A serves as the control, depicting stably tracked kiwifruits. Case B represents kiwifruits that were not tracked stably because they are false positives in some frames. Case C showcases kiwifruits initially obscured by leaves and finally counted due to motion perspective. Case D demonstrates the elimination of sporadic FPs. No overlap between the two containers, which is designed for bi-directional recording and filtered out the unstable tracked object.

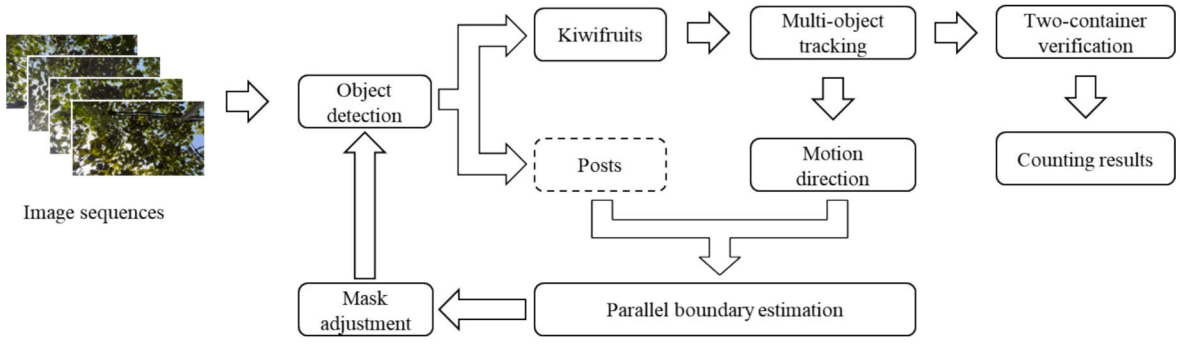


Fig. 7. Row-based counting workflow for kiwifruit orchards. A dashed box indicates the absence of support-post instances in the corresponding frames.

model size and inference time of kiwifruit detection were considered in this study. To facilitate the testing and comparison of different YOLO models, an interface was designed to switch models and manage hyperparameters. The hyperparameters include the intersection over union (IoU) threshold for non-maximum suppression, confidence threshold, operations precise and input size. All models were trained on constructed datasets with identical predefined hyperparameters to ensure consistency and comparability.

The YOLOv5 has undergone a longer period of development, and its stability has constituted the foundation for numerous researches, despite the release of newer detectors with enhanced features (Fu et al., 2024; Xiao et al., 2024). In the comparison among various model scales, YOLOv5 was employed to compare the impact of parameters, model size, inference time and *AP*. This comparison is relevant across different YOLO frameworks because varying scales within each framework are designed by modifying the depth of the model backbone.

It is generally believed that the performance of various models is evaluated on public large datasets. Consequently, an additional experiment was conducted to compare the performance of several popular YOLO frameworks, including YOLOv5, YOLOv6, YOLOv7 and YOLOv8. For consistency, medium-scale models (excluding YOLOv7) were trained and evaluated in this experiment. Within the YOLOv7 frameworks evaluated, the YOLOv7 model was benchmarked against medium-scale models from other frameworks to ensure a fair and uniform comparison.

2.3. *Fruit association based on trackers*

The multi-object tracking system assigns distinct tracking identities (track-IDs) to individual fruits enabling continuous monitoring of their movements, dimensions, and physical attributes until they disappear from the frame. DeepSORT integrates motion and appearance information by combining the marginal distance of the target frame and the cosine distance of the features. It uses a deep neural network to extract characteristics and calculates the similarity matrix of neighboring frames and applies a filtering algorithm with an 8-dimensional vector in the image coordinate system. ByteTrack adopts a lightweight model to reduce the consumption of computational resources, ensuring real-time tracking with stable performance. Unlike traditional approaches that discard low-confidence detection boxes, ByteTrack utilizes the similarities among tracklets to maintain continuity, which is beneficial for mitigating interruptions caused by mutual occlusion and overlap that often occur between fruits and leaves.

Efficient tracking is necessary for kiwifruit counting due to the high density and similar shape of the fruits. The latest track-ID assigned in a stable state is usually considered to be the fruit count. However, due to limited video stabilization, false positives (FPs) resulting from sporadic detector responses are also registered. As Fig. 5 shows, track-IDs were occasionally over-assigned and switched in some frames, resulting in inaccurate fruit counts. To address these challenges, the study compared the inference speed and counting accuracy of DeepSORT and ByteTrack

Table 1
The specifications details of hardware and software.

Item	Training platform	Deploying platform	Smartphone
CPU	Intel Xeon E5-2650v4 Intel Xeon E5-2650v4	Intel Core i5-12400F	Qualcomm Snapdragon 855
GPU	NVIDIA Tesla P100 (16 GB) NVIDIA Tesla P100 (16 GB)	NVIDIA GeForce RTX 3060 (12 GB)	Adreno 640
Memory	512G	32G	8G
OS	Ubuntu 16.04 LTS	Ubuntu 22.04 LTS	ColorOS V12.1
Python	3.7.6	3.7.16	–
PyTorch	1.7.1	1.8.1	–
CUDA	10.1	11.1	–
cuDNN	7.6.5	8.6.5	–

based on YOLOv5m detections. A custom script was developed to manage tracker parameters and facilitate switching between the two methods. The latest coordinates of the detections were matched across consecutive frames and cached, providing data to resolve the issue of over-assigned track-IDs and enhance counting accuracy.

2.4. Fruit counting method

2.4.1. Unstable results elimination

The TCV method was proposed to address the issue of over-assigned track-IDs during tracking. Some assigned track-IDs were registered by sporadic FPs, therefore failed to produce stable and continuous tracklets. The TCV method operates by tracking kiwifruits as they sequentially pass through two predefined virtual containers, as depicted in Fig. 6. A kiwifruit enters the first container, transitions into the second, and is then verified and counted if its trajectory remains stable and consistent throughout this process. The two containers serve as an effective filter mechanism to eliminate FPs that caused by irregular detection responses, ensuring that only real kiwifruits meeting the expected movement criteria are included in the final count.

2.4.2. Row-based counting implementation

A detection region adaptation was introduced to address the issue of over-counting kiwifruits in neighboring rows. This method relies on feedback from the previous frame to refine the region of interest (RoI) and comprises two mechanisms to handle scenarios with and without detected support-posts, as shown in Fig. 7. With support-posts are detected, the bounding boxes of kiwifruit are divided into two groups according to their horizontal coordinates, and virtual parallel boundaries are established to delineate the RoI. The angles of the boundaries are determined by averaging the relative motion direction of the kiwifruits, which serves as a more reliable indicator than the orientation of the support-posts, reducing the impact of random errors. In cases where support-posts are not visible, the RoI mask dynamically adapts by progressively shrinking horizontally while maintaining a distance from estimated boundaries. This ensures that the system continues to function accurately until the support-posts reappear in subsequent frames. By continuously adjusting the mask based on these mechanisms, the detection region adaptation method effectively minimizes over-counting caused by overlapping rows while maintaining robustness under various orchard conditions.

2.5. Application development

A counting platform incorporating the proposed pipeline has been developed to facilitate practical applications based on a C/S architecture. This platform reduces the performance requirements for mobile devices by reallocating complex computations to servers. The platform operates within a Local Area Network (LAN) and establishes communication between the client and server ends via Hypertext Transfer Protocol (HTTP). Users can upload required videos by interacting with the client to the server, which subsequently sends the counting results back after processing. By decentralizing complex computations, this architecture ensures efficient processing while maintaining a user-friendly interface for orchard operators and researchers alike.

2.5.1. Client designment

The client application handles user interactions and can be functionally divided into three parts: media selection, data communication, and result feedback. With improvements in rendering, memory management, and scalability of Webviews, the applications that incorporate web features balance user experience and development efficiency. In this study, the application was implemented using Vue, which is a framework for building user interfaces based on HTML, CSS, and JavaScript. The development environment was based on Microsoft Windows 10 and HBuilderX 3.6.4, and the application was debugged on Microsoft Edge 120.0.2210.144. Furthermore, the application passed functionality tests on OnePlus 7 Pro with Android 12, the specifications of smartphone are listed in Table 1.

The main interface is divided into two sections: a function page and an information page, which can be switched at the tab bar. As depicted in Fig. 8, the function page features a prominent component for uploading media files from the camera or media picker, while the floating action button contains extended functions such as history. The information page includes the version code, guidance, and user feedback. Once the processing results are received, the client navigates to the result page, which can also be accessed through the history page. The layout of the result page is vertically divided into two parts. As shown in Fig. 8c, the processed video is placed in the upper part, while the corresponding count is displayed in the lower part.

2.5.2. Server implementation

The server is built using the Flask web framework, which provides the essentials for web development without unnecessary layers and allows seamless handling of client requests and interactions. It is designed to efficiently handle POST requests from client applications, facilitating communication between the client and server. Upon receiving media files from clients, the server invokes the pipeline with the received media files to perform counting. When the counting is completed, the server sends the processed video and corresponding count data back to the client. Additionally, these results are stored on the server, enabling users to access previously processed files via the history page. In this study, the service was deployed on a PC and its detailed specifications are shown in Table 1.

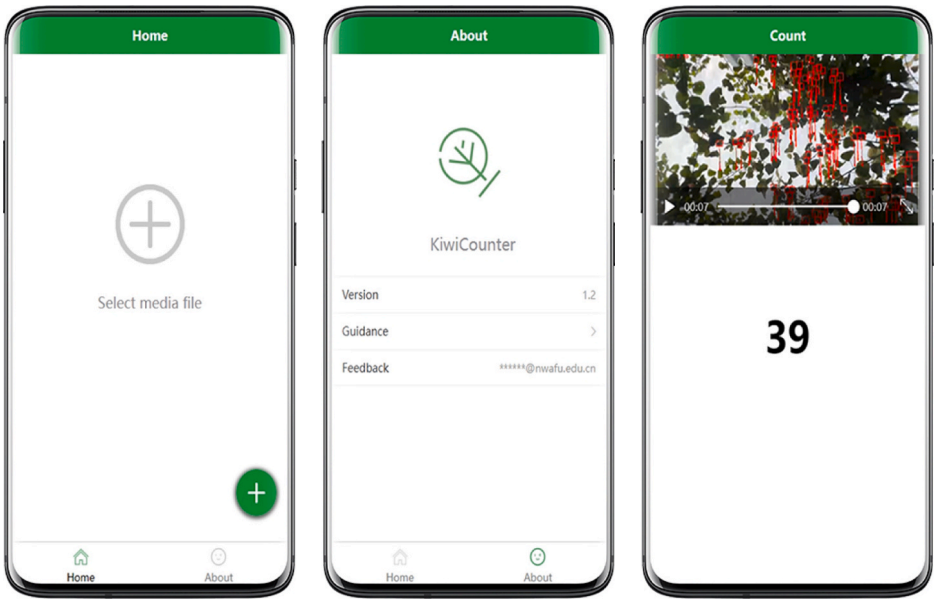
2.6. Experimental details

Experiments were structured into two groups to explore the performance of kiwifruit detection models. The first group focused on comparing performance across different frameworks, and the second group assessed the impact of varying model sizes. All non-quantized models were saved as full-precision floating-point (FP32). The specifications of the training platform utilized for these experiments are detailed in Table 1, while the initial hyperparameters are outlined in Table 2.

2.7. Evaluation metrics

2.7.1. Detection metrics

The trained detection models are evaluated with inference speed and *AP*s. Inference speed refers to the average number of images the model can infer per second, depending on the model parameters, computational device, and input size of the images. The AP_t, as defined by Eq. (1), provides a quantifiable metric that measures the accuracy of model in



(a) The function page of main interface **(b) The information page of main interface** **(c) Result display page of row-based kiwifruit counting**

Fig. 8. The interfaces of client application.

Table 2
Initial hyperparameters.

Item	Value
Initial learning rate	1.0×10^{-3}
Optimizer	Stochastic gradient descent
Scheduler	CosineAnnealingLR
Input size	768 × 768
Batch size	4
Epochs	300
Mosaic probability	1.0
Half-precision	False

both classifying and locating objects at a given IoU threshold t (default $t = 0.5$) in images. $P(r)$ is the precision at a given recall level r while t denotes the preset IoU threshold. Precision (P), as calculated in Eq. (2), indicates the accuracy of positive predictions made by the model. And recall (r) assesses the ability to identify all positive instances, which is calculated in Eq. (3). Microsoft proposed a stricter definition, outlined in Eq. (4), which extends the evaluation by averaging precision across multiple IoU thresholds.

$$AP_t = \int_0^1 P(r)\mathrm{d}r \tag{1}$$

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

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

$$AP_{0.5:0.95} = \frac{1}{10}\sum_{t=0.5}^{0.95} AP_t \tag{4}$$

where TP (True positives) represents the number of correctly predicted positive instances, FP indicates the number of instances that were predicted to be positive but were not, FN (False negatives) refers to the number of positive instances that were not identified by the model, and $AP_{0.5:0.95}$ takes the computation with t ranging from 0.5 to 0.95, with an increment of 0.05.

2.7.2. Tracking and counting metrics

Although several studies have reported fruit counts from video sequences (Wang et al., 2019; Wu et al., 2023), direct comparisons of accuracy continue to pose challenges due to inconsistencies in the data and varying purposes. Moreover, it is challenging to assess row boundaries as these boundaries are invisible and subjectively defined, introducing potential variability and uncertainty into the results.

The performance of the fruit counting is evaluated using the accuracy (Acc) metric, as defined in Eq. (5). This metric measures the ratio of correctly counted kiwifruits to the total number of visible fruits in the 11 test videos. The test videos were labeled as Video i ($i = 1, 2, \ldots, 11$) in accordance with the order of video duration, with Video 1 representing the shortest video and Video 11 representing the longest test video. Additionally, row-based kiwifruit counting accuracy Acc_r was calculated using Eq. (6) to evaluate row-based kiwifruit counting.

$$Acc = 1 - \frac{|\widehat{y} - y|}{\widehat{y} + y} \tag{5}$$

$$Acc_r = 1 - \frac{|\widehat{y} - y_r|}{\widehat{y} + y_r} \tag{6}$$

where y represents the ground truth (GT) for all visible fruit counts while y_r denotes the row-based GT, excluding fruits in neighboring rows, and $\widehat{y}$ represents the algorithm counts. The R-square (R^2), which is defined by Eq. (7), is employed to assess the overall error between the fruit counts and GT.

$$R^2 = \frac{\sum_{i=1}^{11}(y_i - \overline{y})^2}{\sum_{i=1}^{11}(y_i - \widehat{y})^2} \tag{7}$$

where y_i represents the algorithm counts in Video i, $\overline{y}$ denotes the mean of GTs in the 11 test videos.

3. Results and discussion

3.1. Performance of detection models

The medium-scale YOLOv5 model showed great performance on

Table 3
Results of YOLOv5 models in different scales.

Model scale	Params/M	Model size/MB	Inference time/ms	Kiwifruit		Support-post	
				$AP_{0.5}$	$AP_{0.5:0.95}$	$AP_{0.5}$	$AP_{0.5:0.95}$
Nano	**1.9**	**15.7**	**5.4**	0.991	0.833	0.864	0.659
Small	7.2	58.6	6.2	**0.993**	0.828	0.879	0.633
Medium	21.2	170.5	12.3	0.991	**0.864**	0.895	0.727
Large	46.5	373.7	19.1	**0.993**	0.846	**0.929**	**0.787**

Table 4
Comparison of detection frameworks with medium-scale models.

Model scale	Params/M	Model size/MB	Inference time/ms	Kiwifruit		Support-post	
				$AP_{0.5}$	$AP_{0.5:0.95}$	$AP_{0.5}$	$AP_{0.5:0.95}$
YOLOv5m	**21.2**	**170.5**	12.3	0.991	0.864	0.895	0.727
YOLOv6m	34.3	304.8	21.0	0.909	0.730	**0.995**	**0.790**
YOLOv7	36.9	301.8	**11.9**	**0.996**	0.853	0.928	0.750
YOLOv8m	25.9	207.9	16.1	0.983	**0.878**	0.942	0.786



(a) Overexposure with transition from a tight foliage to a sparse foliage.

(b) Fruit mixing into the background in low-contrast imaging.

(c) Motion blur during video recording.

Fig. 9. Kiwifruits that are difficult to detected.

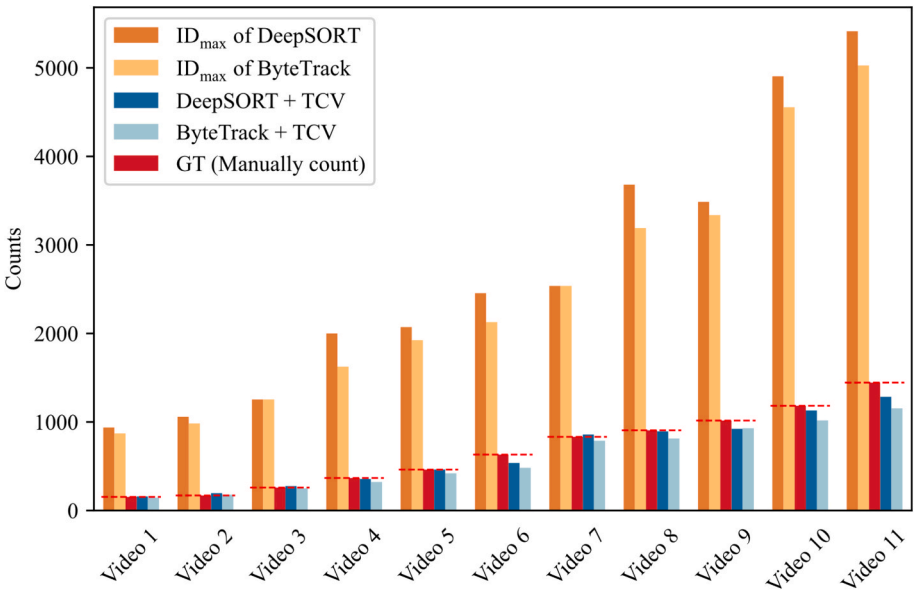


Fig. 10. Kiwifruit counts using different methods. The red dashed lines represent *GT* of videos.

kiwifruit detection, as it strikes a balance between *AP*s and inference time. The *AP*s and inference time of different scale models of YOLOv5 are summarized in Table 3, with the best scores highlighted in bold. As the number of parameters in the feature extraction network increased, the $AP_{0.5}$ of the support-post exhibited an upward trend, which can be attributed to the stronger feature extraction capability associated with a higher parameter count. However, the $AP_{0.5:0.95}$ is more important than the $AP_{0.5}$ in kiwifruit detection, as it evaluates fruits with varying sizes with higher IoU threshold. Although the nano-scale model achieved the fastest inference time and required the smallest model size, its $AP_{0.5:0.95}$

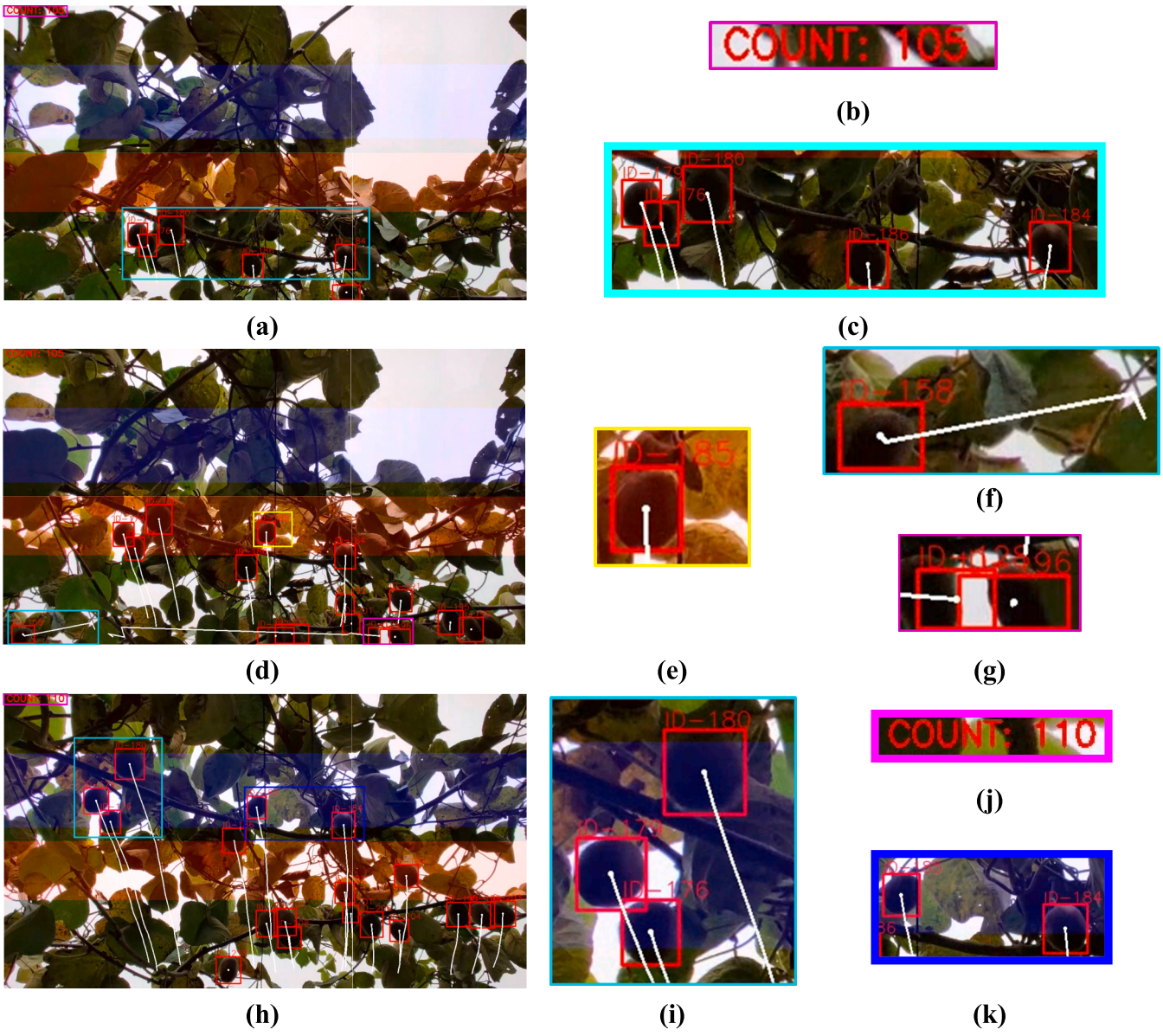


Fig. 11. Examples of TCV method counting, where (a), (d) and (h) are the 241st, 248th and 258th frames in Video 2, aligning with the Stage 1, Stage 2 and Stage 3 in Fig. 6; (b) and (c), (e), (f) and (g), (i), (j) and (k) are the enlarged views of the rectangles with the corresponding colors in (a), (d) and (h); (c) shows the kiwifruits entering the first verification container, while (i) and (k) shows the kiwifruits entering the second verification container; (e) represents the situation that fruits lost detection temporarily, which is the case B in Fig. 6; (f) and (g) represents the ID switches in fruit tracking, which made the counts by direct ID assignment counting unfeasible; (b) and (j) are the count indicators before and after the kiwifruits counting.

Table 5
Comparison of inference speed and counting accuracy with different methods.

Tracker	Method	Inference speed / Frame per second (FPS)	*Acc* /%
DeepSORT	Assigned track-ID	5.80	37.59 %
	Assigned track-ID + TCV	5.32	96.65 %
ByteTrack	Assigned track-ID	23.67	40.12 %
	Assigned track-ID + TCV	22.39	94.20 %

on kiwifruit and support-post were 3.6 % and 10.3 % lower than those of the medium-scale model, respectively. While the small-scale model was 49.6 % faster than the medium-scale model in terms of inference time, its $AP_{0.5:0.95}$ on kiwifruit and support-posts decreased by 4.3 % and 14.8 %, respectively. It is also noticed that the medium-scale model not only achieved the best $AP_{0.5:0.95}$ on kiwifruit detection, but also achieved a 55.3 % faster inference time than the large-scale model. This balance between detection accuracy and computational efficiency makes the medium-scale model the most suitable choice for the task.

Given the need for continuous inference during fruit tracking, YOLOv5m demonstrated a better balance of performance and stability. As shown in Table 4, the YOLOv5m showed a better balance with the best scores marked in bold. It achieved an $AP_{0.5:0.95}$ of 0.864 for kiwifruit detection with fewer parameters (21.2 M) and a faster inference time (12.3 ms). Although YOLOv8m achieved the $AP_{0.5:0.95}$ of 1.6 % and 8.1 % higher than YOLOv5m in kiwifruit detection and support-post detection, respectively, it also showed a 30.9 % increase in inference time. This suggests that YOLOv8m can offer higher precision when processing time is less of a constraint. YOLOv5m and YOLOv7 demonstrated the fastest inference times, both under 13 ms, which meets the real-time requirements for kiwifruit detection. It should be noted that

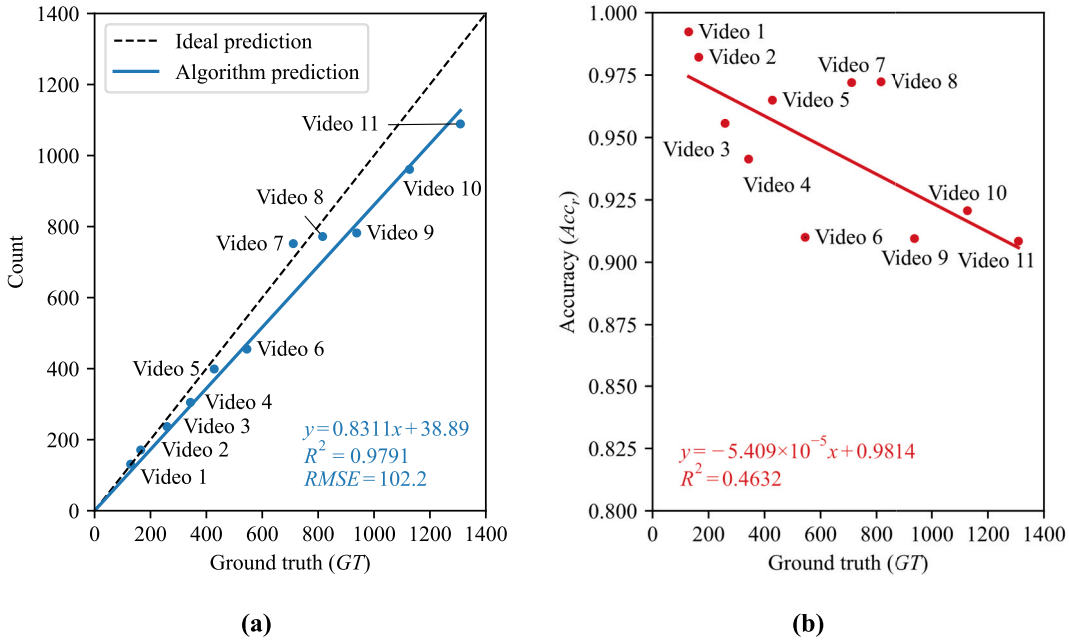


Fig. 12. The result of row-based kiwifruit counts and accuracy. The blue scatter represents the row-based fruit counts based on the ByteTrack and TCV method while the red scatter indicates the accuracy of the aforementioned counts. The solid lines plotted in the same color as the scatters represent the corresponding linear regressions. The black dashed line represents the ideal prediction.

the YOLOv7 model has occupied 131.3 M more storage and 15.7 M more parameters than YOLOv5m model. Despite the use of. strategies like decoupled heads, YOLOv6′s performance in kiwifruit detection was underwhelming. Ultimately, the results indicated that YOLOv5m remained the preferred choice for kiwifruit detection. It exhibits a superior balance between *AP* and inference time in small object detection while maintaining a smaller model size (Lawal et al., 2023; Li et al., 2024b; Zhao et al., 2024).

However, there are some limitations that impact detection and counting accuracy of kiwifruit. For instance, abrupt changes in the surrounding environment and untimely adjustments of imaging parameters during the recording process lead to irreversible loss of image information, as shown in Fig. 9a and Fig. 9b. Motion blur caused by excessive exposure time makes the fruit unrecognizable, as depicted in Fig. 9c. This not only impairs the ability to detect kiwifruit, but also renders manual annotation more challenging. These can be attributed to several factors inherent to the video recording process using smartphone cameras. Moreover, the unbalanced distribution of instances between kiwifruit and support-post may degrade the performance of bounding box regression (Gong et al., 2023; Lee and Ahn, 2023). Additionally, the center drift of bounding box for the support-post is exacerbated due to partial obscuration caused by detection region adaptation, which affect the estimation of row boundary. Finally, detection-based methods face challenges in accurately counting objects in high density scenarios (Fan et al., 2022). These challenges show the shortage at data augmentation and the model robustness with environmental variability.

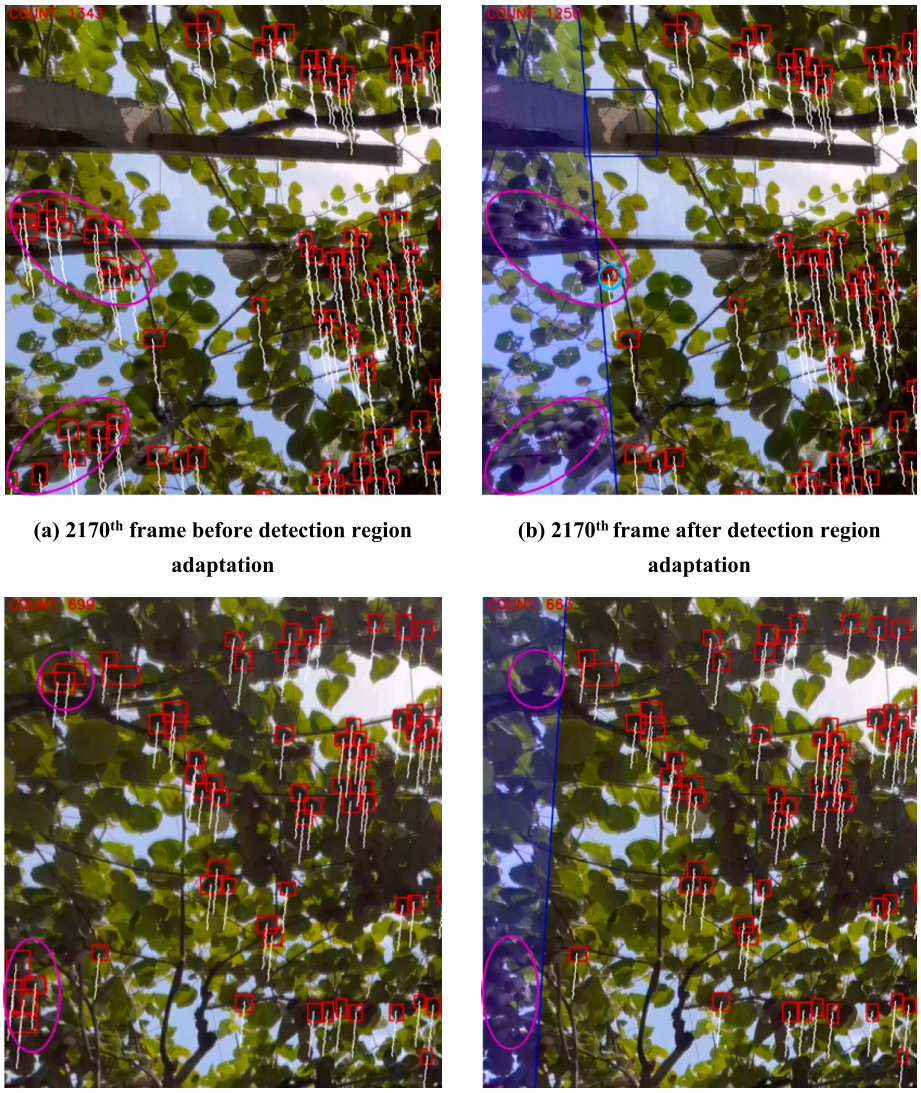
In future investigations, it would be worthwhile to explore from various perspectives. Although some researchers use more detailed data, such as depth information and infrared imaging, to detect kiwifruits (Li et al., 2024a; Liu et al., 2020), these approaches are impractical for smartphone applications. Therefore, there is a pressing need to develop robust data augmentation techniques to address the extreme cases beyond collecting more balanced data. This advancement would enhance the continuity of fruit tracking and mitigate over-estimation. Moreover, more efficient methods exist for high density scenarios to estimate the fruit counts based on the density map estimation, which might provide more detailed results.

3.2. Performance of improved kiwifruit video counting

The TCV method has demonstrated its superiority in counting fruit from videos recorded using smartphones. As illustrated in Fig. 10, the prediction results closely aligned with the *GT* when the TCV method was used, while the results from direct ID assignment counting exhibited noticeable deviations. Moreover, the ID counting method based on both DeepSORT and ByteTrack exhibited a considerable disadvantage, with their maximum track-ID being 2.17 and 2.39 times higher than *GT*, respectively. The reported R^2 values increased by 2.44 % when using TCV method. As depicted in Fig. 11, the TCV method produced less but more accurate fruit counts than the ID counting method, efficiently mitigating over-estimation. As shown in Table 5, the *Acc* based on DeepSORT and ByteTrack increased by 59.06 % and 54.08 %, respectively. These results highlight excellent performance of the TCV method and demonstrate the feasibility of row-based fruit counting.

The TCV method based on ByteTrack showed promising results in accelerating the fruit matching while maintaining a high counting accuracy. As shown in Table 5, the TCV method did not have a significant impact on inference speed. This is due to the limited length of the queue, where the coordinates of the tracked fruit were temporarily stored to discard unnecessary information. Additionally, ByteTrack-based methods were approximately 3.14 times faster than DeepSORT-based methods, and the *Acc* of TCV method based on ByteTrack was 2.45 % less than DeepSORT. Consequently, the integration of TCV method and ByteTrack provides a considerable advantage for real-time counting in orchards.

Although most of the over-estimation in kiwifruit counting was effectively eliminated, several challenges remain. Firstly, the TCV method is inevitably influenced by tracking, and some studies also reported poor accuracy in dense pedestrians tracking, indicating the limitation of current tracking methods in complex and variable backgrounds (Wang et al., 2022). Fortunately, improvements in tracking have shown positive effects in eliminating over-estimation in apple orchards (Wu et al., 2023; Gao et al., 2022). It was also reported that the kiwifruit counts with the direct ID assignment achieved a mean accuracy of 86.18 % when videos were captured closer to the canopies (Guo et al., 2023); however, direct comparison across these studies remains



(a) 2170th frame before detection region adaptation

(b) 2170th frame after detection region adaptation

(c) 1200th frame before detection region adaptation

(d) 1200th frame after detection region adaptation

Fig. 13. Left half frames of examples in row-based kiwifruit counting before and after detection region adaptation in Video 10, where the detected kiwifruits are all marked with red boxes, the white trails indicate the locations they have passed for last 30 frames, the blue line is the estimated boundary after the linearly translation. The kiwifruits falling within the blue polygon (encircled in pink) indicate that they were judged to be outside the boundaries, and the kiwifruits encircled in cyan indicate examples of failed eliminating. The number in the upper left corner indicated the kiwifruit counts as of the current frame; (a) and (b) are the Results of kiwifruit counting with the posts in view; (c) and (d) are the results of kiwifruit counting without posts in view.

challenging due to discrepancies in data collection and research focus. The distinct kiwifruit appearance features and stable relative motion facilitated effective matching across successive frames. In addition, some kiwifruits were filtered out by the TCV method due to visibility lost or blurriness during recording, which was also a disturbing problem in the pear counting that employed RoI line-based (also called line-of-interest) counting (Parico and Ahamed, 2021).

The challenges associated with fruit counting in complex orchard environments have necessitated a comprehensive investigation in future. It is imperative that the containers used in the TCV method should be configured more sensibly to prevent count loss. Also, the effective and stable fruit tracking in complex agricultural scenes remains an intriguing topic warranting further exploration. Moreover. it is also recommended that future research focuses on temporal optical flow analysis in order to resolve FPs by overlapping fruits (Fang et al., 2020), which may help improve the stability of tracking-by-detection methods.

3.3. Performance of row-based kiwifruit counting

The neighboring kiwifruits were hidden by masks during detection, which provided good performance in row-based fruit counting. As shown in Fig. 12a, a linear regression was performed between the counts obtained from the proposed pipeline and the *GT*. The Acc_r exceeds 90 % on the 11 test videos of varying lengths, as demonstrated in Fig. 12b, which indicates that the boundary estimation approximately aligned with the manual annotation. The R^2 associated with the row-based kiwifruit counts was 0.9791, showcasing its stability in row-based kiwifruit counting (Gao et al., 2022; Massah et al., 2021; Xia et al., 2022). However, the study revealed a decline in counting accuracy as the *GT* increased, which is associated with the increased bias in the fitted line of algorithm predictions. A probable explanation is the cumulative effect of errors caused by the constraints both the TCV method and detection region adaptation.

The promising results in row-based fruit counting can be attributed

to the reliable estimation of boundaries and detection region adaptation. As shown in the Fig. 13, the left half of the video frames is displayed for an intuitive comparison. It is worth noting that the estimated boundaries are parallel to the general direction of kiwifruit motion. Fig. 13b shows an example of detection region adaptation guided by support-post. The bounding box did not completely encompass the support-post because the mask generated in the previous frame also made the area outside the row invisible to the detector. Therefore, a linear translation was applied to the.

center of the support-post bounding box to avoid the estimated boundaries getting inward, which allowed for continuous row-based counting in frames without support-post detection.

Despite the promising results, several challenges were identified. The decline in counting accuracy as the *GT* increased, along with the observed bias in row boundary estimation, points to areas for improvement. Although the *GT* of row-based counts may exhibit a degree of subjectivity when counting manually, it is evident that a small number of fruits encircled in cyan, as illustrated in Fig. 13b, were in the neighboring row but were still counted due to the unexpected bias of boundary. This indicates that the estimated boundaries may be conservative or crude. Moreover, it depends on periodically visible support-posts to estimate and adjust the boundaries. But it is difficult to keep boundaries parallel in videos recorded without a preview. The low accuracy of the row-based counts was a consequence of the unstable manner in which the video sequences were captured.

Therefore, future research should focus on developing more sophisticated row boundary estimation methods. User-friendly video sampling should be explored to allow for better control over video quality. Moreover, efforts should also be made to improve the efficiency and flexibility of the counting method, especially in high-density environments.

4. Conclusions

This study proposed an automatic kiwifruit counting pipeline that achieves accurate row-based counting with smartphone-captured videos. YOLOv5m showed its excellent performance in detecting fruits against complex background while maintaining fast inference times, indicating its priority in detecting small-scale objects. The TCV method achieved promising results in video-based fruit counting by filtering out FPs following tracking. Most over-counts from the fruits in neighboring rows were eliminated by detection region adaptation method, suggesting that it is reasonable to estimate the row boundaries with the support-posts.

Although the pipeline showed reliable row-based counting, results were affected as it was difficult to preview during data collection. The study did not consider the impact of route shifts or violent shakes, which may not work on the pipeline. Future research should focus on developing more efficient approaches to achieve row-based kiwifruit counting based on video with a user-friendly capture view. Furthermore, it would be beneficial to investigate utilizing additional field vision data to estimate row boundaries, which could enhance the applicability of the approach in orchards. With further research, the row-based kiwifruit counting pipeline holds significant potential for accurate yield estimation, thereby supporting more informed orchard management decisions.

CRediT authorship contribution statement

Jiwei Zhang: Writing – original draft, Methodology, Investigation, Data curation. **Liguo Jiang:** Writing – review & editing, Investigation, Data curation. **Leilei He:** Writing – review & editing, Methodology. **Zhenchao Wu:** Writing – review & editing, Methodology. **Rui Li:** Writing – review & editing, Methodology, Investigation. **Jinyong Chen:** Writing – review & editing, Methodology, Investigation. **Xiaoxu Sun:** Writing – review & editing, Investigation. **Yunfei Xue:** Writing – review & editing, Data curation. **Anastasia Grecheneva:** Writing – review & editing, Methodology. **Spyros Fountas:** Writing – review & editing, Methodology. **Longsheng Fu:** Writing – review & editing, Supervision, Methodology, Data curation, Conceptualization.

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.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (32171897); Key Research and Development Program of Shaanxi Province, China (2024NC-YBXM-195, 2023JBGS-21); Key Research and Development Program in Henan Province, China (221111111800); National Foreign Expert Project, Ministry of Human Resources and Social Security, China (H20240238, Y20240046).

Data availability

Data will be made available on request.

References

Das, S., Fime, A.A., Siddique, N., Hashem, M.M.A., 2021. Estimation of road boundary for intelligent vehicles based on DeepLabV3+ architecture. IEEE Access 9, 121060–121075. https://doi.org/10.1109/access.2021.3107353.

Fan, Z., Zhang, H., Zhang, Z., Lu, G., Zhang, Y., Wang, Y., 2022. A survey of crowd counting and density estimation based on convolutional neural network. Neurocomputing 472, 224–251. https://doi.org/10.1016/j.neucom.2021.02.103.

Fang, C., Huang, J., Cuan, K., Zhuang, X., Zhang, T., 2020. Comparative study on poultry target tracking algorithms based on a deep regression network. Biosyst. Eng. 190, 176–183. https://doi.org/10.1016/j.biosystemseng.2019.12.002.

Food and Agriculture Organization of the United Nations, 2024. FAOSTAT: Crops and livestock products. https://www.fao.org/faostat/en/#data/QCL.

Fu, L., Liu, Z., Majeed, Y., Cui, Y., 2018. Kiwifruit yield estimation using image processing by an Android mobile phone. IFAC-PapersOnLine 51, 185–190. https://doi.org/10.1016/j.ifacol.2018.08.137.

Fu, X., Zhao, S., Wang, C., Tang, X., Tao, D., Li, G., Jiao, L., Dong, D., 2024. Green fruit detection with a small dataset under a similar color background based on the improved YOLOv5-AT. Foods 13, 1060. https://doi.org/10.3390/foods13071060.

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. Comput. Electron. Agric. 197, 107000. https://doi.org/10.1016/j.compag.2022.107000.

Gong, H., Li, Y., Dong, J., 2023. A dual-balanced network for long-tail distribution object detection. IET Comput. Vis. 17, 565–575. https://doi.org/10.1049/cvi2.12182.

Guo, M., Liu, Y., Li, W., Chen, H., Li, S., Chen, Y., 2023. Real-time production prediction of kiwifruit in orchard based on video tracking algorithm. Trans. Chinese Soc. Agric. Mach. 54 (6), 178–185. https://doi.org/10.6041/j.issn.1000-1298.2023.06.018.

He, L., Fang, W., Zhao, G., Wu, Z., Fu, L., Li, R., Majeed, Y., Dhupia, J., 2022a. 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.

He, L., Wu, F., Du, X., Zhang, G., 2022b. Cascade-SORT: a robust fruit counting approach using multiple features cascade matching. Comput. Electron. Agric. 200, 107223. https://doi.org/10.1016/j.compag.2022.107223.

Lawal, O.M., Zhu, S., Cheng, K., Liu, C., 2023. A simplified network topology for fruit detection, counting and mobile-phone deployment. PLoS One 18, 1–15. https://doi.org/10.1371/journal.pone.0292600.

Lee, H., Ahn, S., 2023. Improving the performance of object detection by preserving balanced class distribution. Mathematics 11 (4460), 1–14. https://doi.org/10.3390/math11214460.

Li, K., Xiong, H., Yu, D., Liu, J., Guo, Y., Wang, J., 2022. An end-to-end multi-task learning model for drivable road detection via edge refinement and geometric deformation. IEEE Trans. Intell. Transp. Syst. 23, 8641–8651. https://doi.org/10.1109/TITS.2021.3084058.

Li, L., He, Z., Li, K., Ding, X., Li, H., Gong, W., 2024a. Object detection and spatial positioning of kiwifruits in a wide-field complex environment. Comput. Electron. Agric. 223, 109102. https://doi.org/10.1016/j.compag.2024.109102.

Li, T., Chen, Q., Zhang, X., Ding, S., Wang, X., Mu, J., 2024b. PeachYOLO: A lightweight algorithm for peach detection in complex orchard environments. IEEE Access 12, 96220–96230. https://doi.org/10.1109/access.2024.3411644.

Li, X., Fang, X., Wang, C., Zhang, W., 2015. Lane detection and tracking using a parallel-snake approach. J. Intell. Robot. Syst. Theory Appl. 77, 597–609. https://doi.org/10.1007/s10846-014-0075-0.

Liang, H., Song, H., Li, H., Dai, Z., 2020. Vehicle counting system using deep learning and multi-object tracking methods. Transp. Res. Rec. 2674, 114–128. https://doi.org/10.1177/0361198120912742.

Lin, H., Yuan, Z., He, B., Kuai, X., Li, X., Guo, R., 2022. A deep learning framework for video-based behicle counting. Front. Phys. 10, 1–11. https://doi.org/10.3389/fphy.2022.829734.

Liu, Z., Wu, J., Fu, L., Majeed, Y., Feng, Y., Li, R., Cui, Y., 2020. Improved kiwifruit detection using pre-trained VGG16 with RGB and NIR information fusion. IEEE Access 8, 2327–2336. https://doi.org/10.1109/access.2019.2962513.

Massah, J., Asefpour Vakilian, K., Shabanian, M., Shariatmadari, S.M., 2021. Design, development, and performance evaluation of a robot for yield estimation of kiwifruit. Comput. Electron. Agric. 185, 106132. https://doi.org/10.1016/j.compag.2021.106132.

Parico, A.I.B., Ahamed, T., 2021. Real time pear fruit detection and counting using YOLOv4 models and Deep SORT. Sensors 21, 4803. https://doi.org/10.3390/s21144803.

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

Tauber, M., Gollan, B., Schmittner, C., Knopf, P., 2023. Passive precision farming reshapes the agricultural sector. Computer 56, 120–124. https://doi.org/10.1109/MC.2022.3218388.

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

Wang, W., Chang, X., Yang, J., Xu, G., 2022. LiDAR-Based Dense Pedestrian Detection and Tracking. Appl. Sci. 12, 1–18. https://doi.org/10.3390/app12041799.

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.

Williams, H.A.M., Jones, M.H., Nejati, M., Seabright, M.J., Bell, J., Penhall, N.D., Barnett, J.J., Duke, M.D., Scarfe, A.J., Ahn, H.S., Lim, J.Y., MacDonald, B.A., 2019. Robotic kiwifruit harvesting using machine vision, convolutional neural networks, and robotic arms. Biosyst. Eng. 181, 140–156. https://doi.org/10.1016/j.biosystemseng.2019.03.007.

Wu, Z., Sun, X., Jiang, H., Gao, F., Li, R., Fu, L., Zhang, D., Fountas, S., 2023. Twice matched fruit counting system: an automatic fruit counting pipeline in modern apple orchard using mutual and secondary matches. Biosyst. Eng. 234, 140–155. https://doi.org/10.1016/j.biosystemseng.2023.09.005.

Xia, X., Chai, X., Zhang, N., Zhang, Z., Sun, Q., Sun, T., 2022. Culling double counting in sequence images for fruit yield estimation. Agronomy 12, 440. https://doi.org/10.3390/agronomy12020440.

Xiao, F., Wang, H., Xu, Y., Shi, Z., 2024. A lightweight detection method for blueberry fruit maturity based on an improved YOLOv5 algorithm. Agric. 14, 36. https://doi.org/10.3390/agriculture14010036.

Xu, H., Zhou, W., Zhu, J., Huang, X., Wang, W., 2017. Vehicle counting based on double virtual lines. Signal, Image Video Process. 11, 905–912. https://doi.org/10.1007/s11760-016-1038-7.

Zhang, X., Li, X., Zhang, B., Zhou, J., Tian, G., Xiong, Y., Gu, B., 2018. Automated robust crop-row detection in maize fields based on position clustering algorithm and shortest path method. Comput. Electron. Agric. 154, 165–175. https://doi.org/10.1016/j.compag.2018.09.014.

Zhao, P., Zhou, W., Na, L., 2024. High-precision object detection network for automate pear picking. Sci. Rep. 14, 1–20. https://doi.org/10.1038/s41598-024-65750-6.

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.