El Akrouchi, M., Mhada, M., Bayad, M., Hawkesford, M. J. and Gerard, B. 2025. AI-based framework for early detection and segmentation of green citrus fruits in orchards. *Smart Agricultural Technology.* 10, p. 100834. https://doi.org/10.1016/j.atech.2025.100834

Smart Agricultural Technology 10 (2025) 100834

Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

AI-based framework for early detection and segmentation of green citrus fruits in orchards

Manal El Akrouchi [a,c,*], Manal Mhada [a], Mohamed Bayad [a], Malcolm J. Hawkesford [b], Bruno Gérard [a]

[a] *College of Agriculture and Environmental Sciences, University Mohammed VI Polytechnic (UM6P), Ben Guerir, Morocco*
[b] *Sustainable Soils and Crops Department, Rothamsted Research, Harpenden, Hertfordshire, United Kingdom*
[c] *School of Collective Intelligence, University Mohammed VI Polytechnic (UM6P), Rabat, Morocco*

ARTICLE INFO

Keywords:
Citrus
Deep learning
Instance segmentation
Cascade Mask R-CNN
Transformers
Yield prediction
Precision agriculture

ABSTRACT

The detection and segmentation of tiny green citrus fruits in dense orchards play a vital role in modern farming, directly influencing yield prediction, resource management, and timely decision-making. This research presents a cutting-edge framework that combines Multiscale Vision Transformers version 2 (MViTv2) with Cascade Mask R-CNN to tackle these challenges effectively. By extending the focus from close-up images to the novel inclusion of full-tree images, the framework enables accurate early-stage detection, segmentation, and counting of citrus fruits in practical orchard settings. Unlike conventional methods, this approach uses a dual-image strategy: close-up images for training and full-tree images—more complex due to dense foliage and small fruits—for testing and real-world applications. To enhance detection accuracy in these detailed, full-tree images, the framework employs an innovative image-slicing method, breaking high-resolution images into smaller parts to capture finer details. The model was tested on a unique dataset featuring citrus orchards of three varieties: Nules grafted on Volka, Sidi Aissa grafted on Volka, and Orogrande grafted on sour orange. Results showed that the MViTv2_L backbone outperformed alternatives, achieving a mean Average Precision (mAP) of 72.97% for bounding boxes and 84.40% for masks. The image-slicing technique further boosted fruit detection in full-tree images, achieving an R^2 value of up to 0.81 for fruit counting. This dual-image method, paired with advanced segmentation and detection technologies, marks a significant step forward for agricultural robotics and precision farming, enabling accurate early-stage fruit detection in real-world orchard environments.

1. Introduction

Advancements in artificial intelligence (AI) and deep learning (DL) have revolutionized computer vision, enabling the development of innovative solutions for complex challenges across diverse domains. AI has facilitated transformative applications in areas such as autonomous driving, medical imaging, and precision agriculture [14,38]. Deep learning, in particular, has enhanced the capabilities of computer vision by enabling algorithms to process and interpret large-scale image data with remarkable accuracy [10,39,32,2,25].

In the context of precision agriculture, these advancements have provided farmers with tools to improve productivity, optimize resource management, and minimize environmental impact [19,35]. Precision agriculture integrates advanced technologies to monitor and manage crop growth, enabling timely decision-making for irrigation, pest control, and harvest planning [16,40]. Among its key applications is the detection and segmentation of crops and fruits, tasks that are critical for yield forecasting and agricultural interventions.

Citrus fruits, as a globally significant crop with over 143 million tons produced annually,[1] present unique challenges for detection. Their small size, unripe green coloration, and the dense foliage in orchards complicate automated recognition and segmentation [3]. The ability to detect and segment citrus fruits at an early stage is pivotal for optimizing resource allocation and yield prediction [31,30]. Despite these

* Corresponding author at: College of Agriculture and Environmental Sciences, University Mohammed VI Polytechnic (UM6P), Ben Guerir, Morocco.
E-mail addresses: manal.elakrouchi@um6p.ma (M. El Akrouchi), manal.mhada@um6p.ma (M. Mhada), mohamed.bayad@um6p.ma (M. Bayad), malcolm.hawkesford@rothamsted.ac.uk (M.J. Hawkesford), bruno.gerard@um6p.ma (B. Gérard).

[1] https://www.fao.org/policy-support/tools-and-publications/resources-details/en/c/1439010/.

https://doi.org/10.1016/j.atech.2025.100834
Received 24 November 2024; Received in revised form 30 January 2025; Accepted 9 February 2025

Available online 12 February 2025

challenges, early detection is critical to improving agricultural practices, as it facilitates timely interventions, including irrigation scheduling and pest management.

Traditional methods for citrus detection have employed image processing and convolutional neural networks (CNNs). While these approaches have demonstrated reasonable performance, they often struggle with real-world orchard environments, where fruit occlusion, varying light conditions, and complex backgrounds dominate [17,26]. Choi et al. [7] and Dorj et al. [8] both developed algorithms for citrus recognition and counting, with Dorj et al. [8] achieving a high correlation coefficient of 0.93 and Choi et al. [7] achieving a 90% correct identification rate. Qin et al. [29] and Lyu et al. [24] both proposed target detection models, with Qin et al. [29] achieving a 90% correct identification rate and Lyu et al. [24] achieving a 98.23% mAP@.5 for green citrus. Chen et al. [5] focused on citrus recognition in different growth periods, achieving a segmentation accuracy of 94.87% for green citrus and 97.08% for yellow citrus. However, most existing methods rely heavily on close-up images, neglecting the complexity of natural orchard environments, where dense foliage and variable lighting conditions obscure fruits. Moreover, the challenges posed by the small size of unripe fruits, their tendency to blend with foliage, and orchard-level variability require more advanced methods capable of operating in real-world conditions [17].

This paper introduces a novel framework to address these challenges by integrating Multiscale Vision Transformers version 2 (MViTv2) with Cascade Mask R-CNN. This framework combines the robust feature extraction capabilities of MViTv2 with the precision of Cascade Mask R-CNN for detection and segmentation tasks. Cascade Mask R-CNN is a state-of-the-art instance segmentation model that progressively refines predictions across multiple stages, ensuring higher accuracy for challenging objects and complex environments [4,13]. It has been successfully applied in fields such as autonomous driving and agricultural monitoring [12,27]. Coupled with vision transformers, which excel at capturing fine-grained details and handling dense prediction tasks, this framework significantly improves detecting and segmenting citrus fruits in real-world conditions [9,36]. MViTv2, a multiscale variant of vision transformers, enhances feature recognition across various scales, making it particularly effective in distinguishing small fruits from foliage in cluttered images.

To enhance the detection of small citrus fruits in full-tree images, the framework incorporates an image-slicing technique that divides high-resolution images into smaller segments, allowing the model to process intricate details more effectively and improve the detection of small fruits obscured by dense foliage. Additionally, a dual-image strategy is introduced: close-up images are used for training, while full-tree images, capturing the complexity of natural orchards, are employed for testing and application. The framework was tested using different backbones and different slicing strategies, ensuring the model is well-trained to handle real-world agricultural scenarios.

The paper is structured as follows: Section 2 describes the data used in this research and offers a detailed account of the proposed framework, including the specific architecture of MViTv2 and Cascade Mask R-CNN. Section 3 presents the results, including an assessment of evaluation metrics, while Section 4 discusses these findings and addresses certain limitations. Finally, Section 5 concludes the paper by summarizing the discoveries and providing suggestions for future work.

2. Material and methods

This study presents an innovative approach to detecting and segmenting tiny green citrus fruits at an early stage. The following flowchart in Fig. 1 comprehensively illustrates the activities integral to the study. The following sections describe the processes in detail.

2.1. Plant materials

The experiment was conducted in a commercial orchard Cap agro (Jnane Rhamna Farm), located 35 km north of Marrakech (Morocco, 52° 26’ 56.004”N, 9° 44’ 24”E). The study was carried out on 11-year-old trees of the three clementine varieties:

1. Nules (*Citrus clementina, Hort ex Tan*) grafted on Volka (*Citrus Volkameriana*): characterized by the early flowering stage, which started on March 1st, 2022, and harvested on November 1st, 2022.
2. Sidi Aissa (*Citrus reticulata Blanco*) grafted on Volka (*Citrus Volkameriana*): characterized by mid-early flowering, which started on March 15th, 2022, and harvested on November 15th, 2022.
3. ‘Orogrande’ grafted on sour orange (*Citrus aurantium L.*): a late-flowering variety that started flowering on April 1st, 2022 and harvested on December 1st, 2022.

The trees have an estimated lifespan of around 25 years, displaying similar growth dynamics, and their flowering phase persists for two months across all varieties. The trees were planted at a spacing of 6 m × 3 m on ridges made from the soil taken from the area between the rows to increase soil depth and improve water drainage in the orchard. These ridges are approximately 30 cm in height and 1.5 m in width. The trees were ferti-irrigated using two lines of drippers for each tree row, one on each side of the row placed 1.0 to 1.2 m away from the tree trunk. The drippers were 1 m apart on the line and had a flow rate of 6 L h-1 dripper-1. Weeds, diseases, and pests were controlled according to local criteria and regulations.

2.2. Data collection

The field measurements were conducted on 15th June 2022, and ground-based images were captured for citrus tree phenotyping. The imaging equipment consisted of a SONY ILCE-5100, a 24.3-megapixel digital camera (6000 × 4000 pixels), and a 35 mm camera lens. In addition, the field imagery was captured under natural lighting conditions using a color checker for accuracy. Images were taken from both sides of each tree using two different protocols. In the first protocol, a camera was placed close to the tree (between 50 cm and 80 cm away) to capture small green citrus fruits. The camera settings were as follows: Focal length: 35 mm, Aperture: f/10.0, ISO: AUTO, and Exposure time: 1/400 s. The ground resolution of the images was approximately between 0.1784 and 0.179 mm per pixel.

For the second protocol, the goal was to capture the entire tree in a single image. This would allow us to address the challenge of detecting and segmenting small, unripe citrus fruits that blend with foliage in complex agricultural environments. To achieve this, the camera was positioned 3 meters from the tree, and the following settings were used: Focal length of 18 mm, Aperture of f/10.0, ISO: AUTO, and Exposure time of 1/400 s. The ground resolution of the images ranged from approximately 0.0348 to 0.0346 cm per pixel.

Figs. 2 present an example of images taken using the two protocols.

Citrus fruits typically grow in clusters along branches, often partially occluded by dense foliage, and their spatial distribution is significantly variable. The fruits’ size and visibility depend on their growth stage, with smaller, unripe fruits often blending with the surrounding greenery. These growth characteristics, combined with environmental factors such as varying light intensities, shadows, and background complexity, make citrus fruit detection particularly challenging in orchard environments.

The dataset used in this study was carefully designed to reflect these real-world growth distributions and environmental conditions while also capturing diversity in citrus varieties. It includes close-up and full-tree images from three distinct citrus varieties. These varieties exhibit differences in fruit size and canopy structure, comprehensively representing the variability found in citrus orchards. This diversity ensures

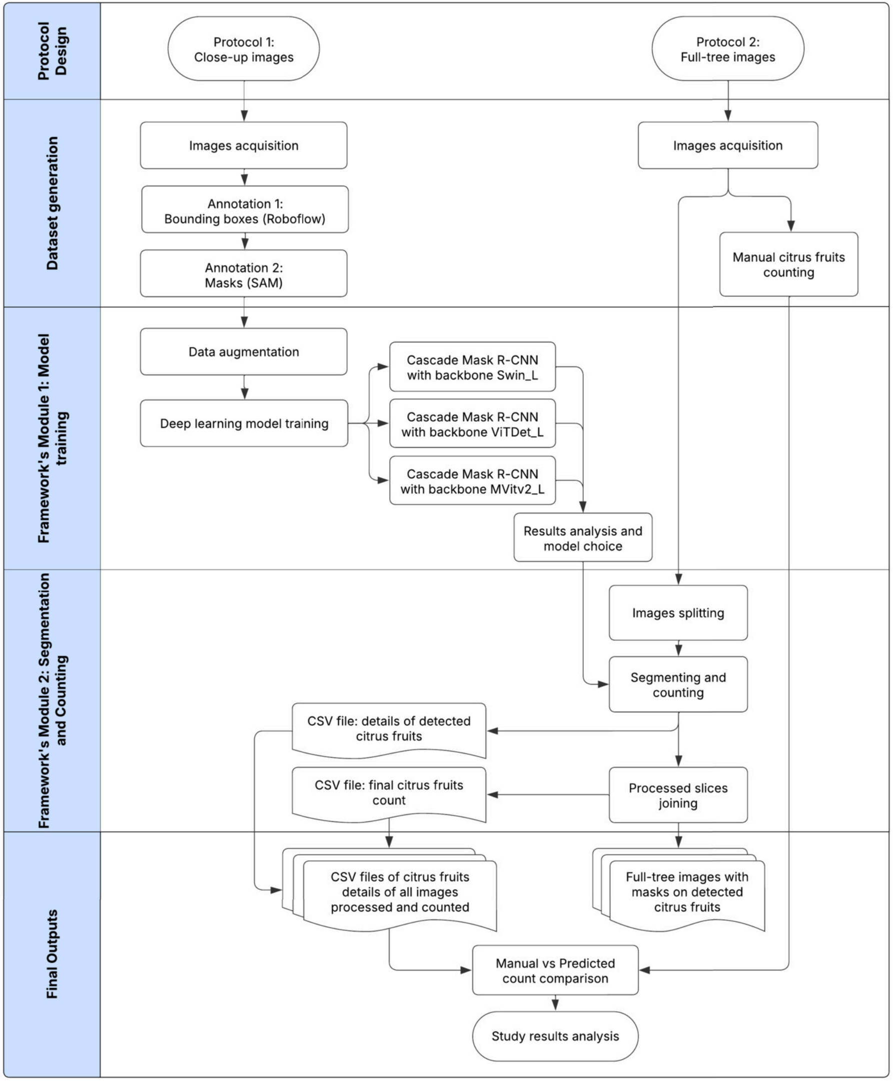
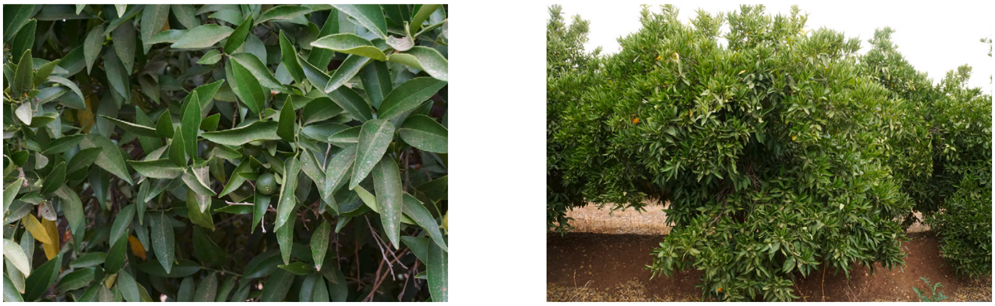

Protocol Design
Dataset generation
Framework's Module 1: Model training
Framework's Module 2: Segmentation and Counting
Final Outputs
Protocol 1: Close-up images
Protocol 2: Full-tree images
Images acquisition
Annotation 1: Bounding boxes (Roboflow)
Annotation 2: Masks (SAM)
Images acquisition
Manual citrus fruits counting
Data augmentation
Deep learning model training
Cascade Mask R-CNN with backbone Swin_L
Cascade Mask R-CNN with backbone ViTDet_L
Cascade Mask R-CNN with backbone MVitv2_L
Results analysis and model choice
Images splitting
Segmenting and counting
CSV file: details of detected citrus fruits
CSV file: final citrus fruits count
Processed slices joining
CSV files of citrus fruits details of all images processed and counted
Full-tree images with masks on detected citrus fruits
Manual vs Predicted count comparison
Study results analysis

Fig. 1. Overall process flowchart of small green citrus fruits detection and segmentation.



(a) Example of the close image.

(b) Example of the full tree image.

Fig. 2. Examples of images collected: **(a)** Using the first protocol. **(b)** Using the second protocol.

the algorithm is trained and tested across various scenarios, improving its robustness and adaptability. Key features of the dataset include:

- Variety-driven variability: By incorporating multiple citrus varieties, the dataset reflects differences in fruit clustering, density, and canopy complexity, allowing the framework to generalize across diverse orchard setups.
- Fruit clustering and variability: The dataset contains images with fruits in dense clusters, isolated instances, and overlapping configurations, testing the algorithm's ability to handle varying spatial distributions.
- Foliage density and occlusion: Images with varying levels of foliage density were included to assess the framework's capability to detect partially and fully occluded fruits.
- Environmental diversity: The dataset captures diverse environmental conditions, including bright sunlight, shaded areas, and transitional lighting, mimicking the variability seen in real-world orchards.

Table 1
Description of the dataset created using the first image taking protocol.

	Total Number of Images	Total Number of Citrus Fruits	Average Citrus Fruits per Image	Minimum Citrus Fruits per Image	Maximum Citrus Fruits per Image
Train Dataset	300	1068	3	1	12
Validation Dataset	69	247	3	1	15
Test Dataset	30	94	3	1	10

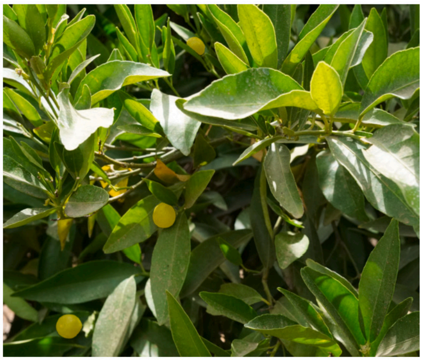


Fig. 3. Annotated image of small green citrus fruits with yellow masks.

2.3. Dataset preparation

The images taken in the first protocol were annotated using the tool Roboflow.[2] A total of 399 unique images representing all varieties were annotated using bounding boxes on each citrus fruit. The initial aim was to develop a model for detecting citrus fruits in full tree images using object detection techniques. However, the model failed to detect the fruits accurately due to their size and color. To improve detection and segmentation, Facebook's "Segment Anything" model [18] was adopted.

The SAM model is a groundbreaking tool that offers both versatility and efficiency. It can accurately segment objects, even in complex scenes, significantly reducing the manual effort and time required for annotation. A range of studies have explored the Segment Anything Model (SAM) applications in various fields. Cheng et al. [6] introduces SAM-Track, a framework for precise object segmentation and tracking in videos, with applications in drone technology, autonomous driving, medical imaging, augmented reality, and biological analysis. Sun et al. [33] demonstrates the potential of SAM in weakly-supervised semantic segmentation, achieving impressive results on PASCAL VOC and MS-COCO datasets. Finally, the 399 images were re-annotated using masks, as presented in Fig. 3, segmenting over 1400 citrus fruits.

All images were thoughtfully combined into a unified dataset to train the model on a broad spectrum of variations. This careful consolidation allowed the model to learn a diverse range of features, thereby enhancing its overall learning experience. The dataset was then organized into separate segments for training, validation, and testing, ensuring a comprehensive framework for effective model training. The resulting dataset was saved in COCO format. A detailed overview of the data distribution is provided in Table 1.

The second protocol is designed to systematically capture full images of trees, resulting in a comprehensive total of 48 images. For each variety, 16 unique high-resolution images are selected, ensuring a thorough representation by photographing both sides of each tree.

2.4. Methodology

The proposed framework is designed to identify, segment, and count small green citrus fruits in their early stage. It comprises two main modules: The first module focuses on training a deep-learning model for detection and segmentation, utilizing annotated images captured through the first image-taking protocol to ensure maximum accuracy. The chosen model is the Cascade Mask R-CNN with the MViTv2_L backbone. The second module is responsible for identifying and counting small green citrus fruits in full tree images, which are taken using the second image-taking protocol. This module comprises three components: Slicing, Segmentation and counting, and Joining. Fig. 4 presents an overview of the proposed framework.

2.4.1. Model's architecture

The framework is based on Cascade Mask R-CNN. With its multi-stage refinement and combination of classification, localization, and segmentation losses, the Cascade R-CNN framework ensures that the model progressively improves its detection and segmentation capabilities. This comprehensive approach allows for high-quality object detection and instance segmentation, addressing the challenges of precise localization and segmentation in complex scenarios. Additionally, selecting the Transformers as the backbone for this model presents a strategic enhancement. Transformers, known for their exceptional performance in capturing long-range dependencies and contextual information, complement the Cascade Mask R-CNN's hierarchical structure. This combination leverages the strengths of both architectures: the transformers provide a robust feature extraction mechanism, enhancing the model's understanding of spatial relationships, while the Cascade Mask R-CNN excels in precise object detection and instance segmentation. The choice was then a powerful transformer called MViTv2.

Multiscale Vision Transformers version 2 (MViTv2) represents a significant advancement in computer vision, especially in tasks requiring nuanced detail recognition. MViTv2, an extension of the initial Multiscale Vision Transformers [9], leverages a hierarchical transformer architecture designed to handle diverse image resolutions effectively. This design enables the model to capture fine-grained details at multiple scales, making it particularly suitable for complex tasks such as detecting and segmenting small or densely packed objects in cluttered scenes [20].

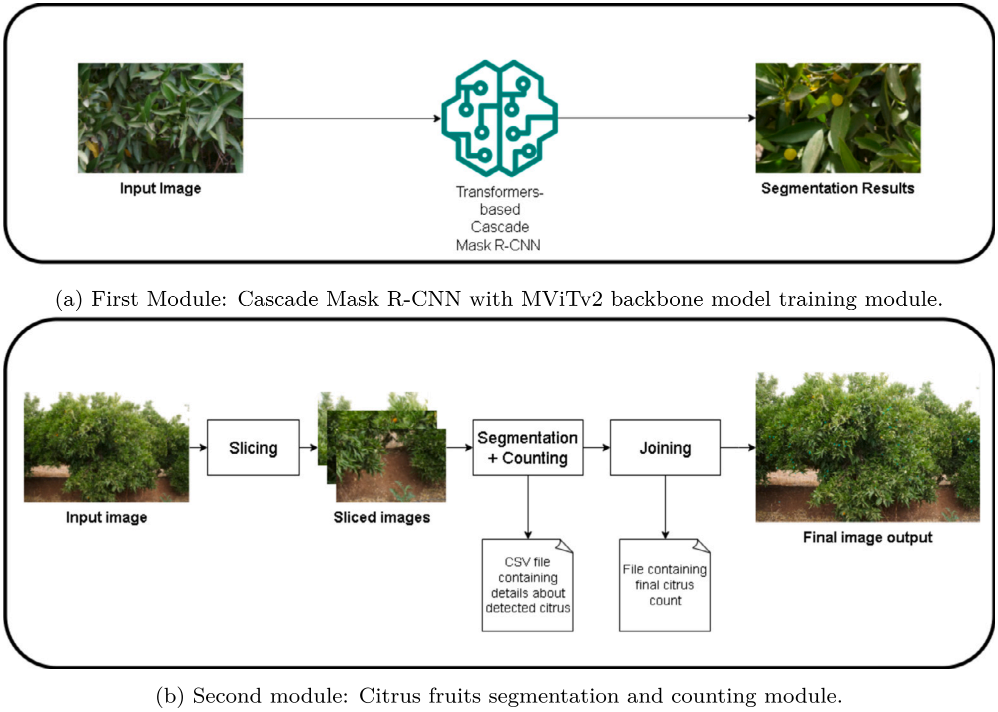
One of the key enhancements in MViTv2 is its improved efficiency in processing high-dimensional data, achieved through optimizations in its attention mechanisms and scaling strategies [20]. Due to the dense coverage of the tree leaves, the MViTv2 L backbone was opted for using stronger large-scale jittering training [11]. Fig. 5 displays the adopted model's architecture.

2.4.2. Model training and evaluation

This study was implemented using Python 3.9 and Pytorch 2.0 framework. All the models were trained in Google Colab A100-SXM4-40GB GPU. The Cascade Mask R-CNN model is implemented using Detectron2, a powerful software system developed by Facebook AI Research (FAIR) [37]. Detectron2 is an upgraded version of Detectron, coded in PyTorch with a more modular design. It can implement advanced algorithms such as Faster R-CNN, Mask R-CNN, RetinaNet, and DensePose. Its heightened flexibility and extensibility have made it FAIR's most popular open-source project. After several trials and errors, the model was trained for 2000 iterations.

Due to the limited availability of datasets, transfer learning has become a popular approach to train deep learning models more efficiently and stably [34]. By leveraging pre-trained MViTv2 features from ImageNet21k, which consists of 21,843 object categories and 14 million images at resolution 224x224, state-of-the-art results have been achieved

[2] https://roboflow.com/.


Input Image
Transformers-based Cascade Mask R-CNN
Segmentation Results
Input image
Slicing
Sliced images
Segmentation + Counting
Joining
Final image output
CSV file containing details about detected citrus
File containing final citrus count

(a) First Module: Cascade Mask R-CNN with MViTv2 backbone model training module.

(b) Second module: Citrus fruits segmentation and counting module.

Fig. 4. Overview of the proposed framework.

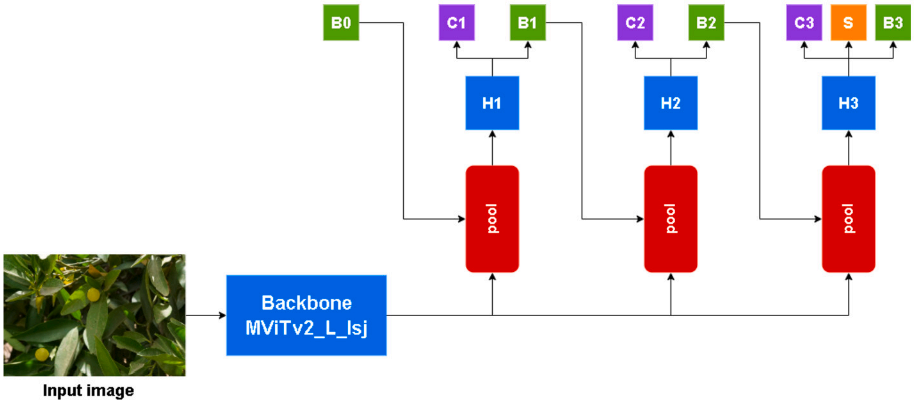


Fig. 5. Schematic diagram of Cascade Mask R-CNN model. “Backbone” is the transformer-based backbone, “pool” is region-wise feature extraction, “H” is the network head, “B” is the bounding box, and “C” is the classification. “B0” refers to proposals in all architectures. “S” denotes a segmentation branch.

in various image processing tasks, ranging from image classification to image captioning. Fine-tuning the pre-trained model’s layers with the labeled citrus fruits image is necessary.

Data augmentation is necessary to improve the dataset for training, as it increases the number of images while maintaining quality [28]. Data augmentation was applied using the defined functions:

- RandomFlip: Flip the image horizontally or vertically with the given probability.
- ResizeScale: Takes target size as input and randomly scales the target size between min_scale of 0.1 and max_scale of 2.0. It then scales the input image to fit inside the scaled target box, keeping the aspect ratio constant.
- FixedSizeCrop: Crop a region out of an image with a fixed crop_size of [1024, 1024].

Hyperparameters play a pivotal role in the training and performance of deep learning models, and Cascade Mask R-CNN is no exception. In Cascade Mask R-CNN, hyperparameters, such as learning rate, batch size, weight decay, and anchor scales, significantly influence the network’s convergence rate, adaptability to the dataset, and detection and segmentation accuracy. Several hyperparameters were fine-tuned in the experiments to better align with this study’s dataset characteristics. The learning rate was set to 0.0001, with a weight decay 0.0001 and AdamW [22] as the optimization method.

The loss calculation for Cascade Mask R-CNN is derived from the Faster R-CNN architecture’s multi-stage extension, Cascade R-CNN [21]. The main objective of Cascade R-CNN is to enhance object detection by progressively refining bounding box predictions through multiple stages. Each stage in the cascade is trained to handle progressively higher Intersection over Union (IoU) thresholds, which helps achieve better object localization.

For each stage t, the total loss is a combination of the classification loss L_{cls} and the localization loss L_{loc}:

$$L_t = L_{\text{cls}}(h_t(x_t), y_t) + \lambda[y_t \geq 1] L_{\text{loc}}(f_t(x_t, b_t), g), \tag{1}$$

where h_t is the classifier, f_t is the regressor, x_t represents the input, y_t is the label under the IoU threshold u_t, b_t is the predicted bounding box, and g is the ground truth bounding box. The indicator function $[y_t \geq 1]$ ensures that the localization loss is only applied to positive samples. The classification loss L_{cls} is typically computed using cross-entropy loss, while the localization loss L_{loc} is computed using the smooth L1 loss on the bounding box coordinates.

In the Cascade Mask R-CNN, an additional segmentation loss L_{seg} is introduced for instance segmentation:

$$L_{\text{seg}}(m_t, s_t), \tag{2}$$

where m_t is the predicted mask and s_t is the ground truth segmentation mask. This segmentation loss is typically computed using binary cross-entropy or a similar pixel-wise loss function. The segmentation branch can be added at the first stage, at the last stage, or each stage of the Cascade R-CNN, and the final mask prediction is obtained from the single or ensemble segmentation branches, depending on the architecture.

The losses from different stages are combined using schemes such as "average" (average) and "decay." In the "average" scheme, the loss of each stage receives an equal weight, whereas, in the "decay" scheme, the loss of each stage is weighted, giving more importance to earlier stages in training. The total loss across all stages is combined to optimize the model progressively. The average scheme for loss combinations was adopted, as originally used in the Detectron2 implementation of Cascade Mask R-CNN.

The predicted segmentation masks in the output images were obtained from the trained Cascade Mask R-CNN and put for further analysis. The aim was to evaluate the effect of the different backbone parts in the Cascade Mask R-CNN mask. This analysis used two metrics: average precision (AP) and IoU. IoU is a crucial metric used to assess segmentation models [42], commonly referred to as Jaccard's Index. This metric quantifies how effectively the model can distinguish objects from their backgrounds in an image.

The Intersection over Union (IoU) between the ground-truth fruit region, A_{gt}, and the predicted fruit region, A_p, was calculated as follows:

$$IoU(A_{gt,Ap}) = \frac{A_{gt} \cap A_p}{A_{gt} \cup A_p} \tag{3}$$

One of the most crucial evaluation indicators for measuring the object detection model's performance is mean average precision (mAP), which can effectively evaluate the locating performance of the model. In order to assess the performance of the model, the official COCO evaluation metrics in Python were employed, including AP50 and AP75, defined as follows:

1. AP at IoU = 0.5 (AP50)This version of the AP metric evaluates average precision when the Intersection over Union (IoU) threshold is set at 0.5. A higher IoU threshold means stricter evaluation criteria and an IoU of 0.5 is commonly used for many detection tasks.
2. AP at IoU = 0.75 (AP75): This is similar to AP50 but uses a more stringent IoU threshold of 0.75, focusing on tighter bounding box matches.

All experiment results were obtained at a threshold of IoU = 0.5. These metrics offer a thorough evaluation of bounding box and mask annotations. However, to guarantee the accuracy of the predicted count, it was cross-referenced with the count determined by an expert. By utilizing both approaches, the models may then be assessed more accurately.

2.4.3. Segmentation and counting module

Due to the size and color of the citrus fruits, the model trained may not be able to detect and segment all fruits. The challenge of diminished model accuracy when analyzing full-tree images for citrus fruit detection was addressed using the slicing strategy. When slicing the image into many parts, the model will treat and analyze each slice as one large single image and try to detect all citrus in it.

In addition, another advantage of the slicing approach is the slice size: the full image will have a large size and may consume more time and energy for detection and counting; however, when slicing, the new image fragments will have reduced size, making it quicker and easier for the model to detect and count the fruits. This division is crucial as it counteracts the issues related to scale and complexity inherent in full-tree images, which often lead to reduced detection accuracy.

By focusing on smaller sections of the image, the model can more effectively apply its detection capabilities, as each segment presents the fruits and foliage in greater detail and less cluttered contexts. This strategy ensures that the vast and varied background of full-tree images does not overshadow the nuances and characteristics of tiny citrus fruits.

When analyzing the literature, the SAHI (Slicing Aided Hyper Inference) framework Akyon et al. [1] was first adopted. This approach is centered around image slicing, where large images of entire trees are methodically divided into smaller, more manageable segments. The SAHI framework relies on slicing the images for analysis based on some parameters that need to be defined to adjust the inference. The concept of sliced inference is basically performing inference over smaller slices of the original image and then merging the sliced predictions on the original image.

SAHI (Slicing Aided Hyper Inference) Akyon et al. [1] is a technology known for its precision in detecting small objects with high accuracy. However, two primary challenges must be addressed: inference time and instance segmentation.

For inference time, SAHI's performance slows significantly when processing high-resolution images, primarily due to the size of the image. The inference time can be substantial, with some reports indicating it can take up to 30 minutes to process a single image. This prolonged processing time is influenced by the need to determine the optimal size for image splits. Larger images require more computational resources and time, making real-time application impractical in its current form.

The second challenge pertains to instance segmentation. SAHI aims to extract comprehensive information and features from detected objects, such as citrus fruits. However, achieving high-quality instance segmentation with SAHI can be difficult. The technology struggles to provide detailed feature extraction, which is crucial for applications requiring precise object identification and classification.

The proposed segmentation and counting module is meticulously designed to deliver accurate results while significantly reducing processing time. This advanced module comprises several integral components, each contributing to the efficiency and precision of the overall process.

The first component is a slicing mechanism that divides the input image into uniformly sized tiles. This tiling approach ensures that each segment is manageable for more precise detection and segmentation. Once the image is divided, the model trained in the previous module is employed to process each tile individually. The segmentation and counting component of the module performs two critical tasks. Initially, it detects and segments the citrus fruits within each tile, drawing masks over each detected fruit to visually delineate them. This visual segmentation is crucial for accurate counting and further feature extraction. Following the segmentation, the module extracts essential features from each detection. These features include the size of each mask, the location coordinates of each detected citrus fruit, and a unique ID for each fruit. All this detailed information is systematically exported into a CSV file, creating a comprehensive dataset for further analysis.

The final step of the module is the 'joining' process. All the tiles, now drawn with masks, are reassembled to form a complete image. This step is crucial for maintaining the visual integrity of the original image, which is now enhanced with detailed segmentation data. The module's tasks culminate in recording the total count of citrus fruits detected in each processed image. This count is then saved into another CSV file, providing an easily accessible record of the fruit count for each image.

3. Results

This study evaluates the proposed framework using two distinct image protocols. The first module focuses on close-up images, where the

Table 2
Results of AP metrics on the testing dataset using different backbones.

Backbone	mAP		AP50		AP75	
	Bbox %	Mask %	Bbox %	Mask %	Bbox %	Mask %
Swin_L	55.863	64.869	88.283	86.203	65.247	73.078
ViTDet_L	68.515	83.849	93.797	92.566	87.399	91.767
MViTv2_L	**72.97**	**84.40**	**98.60**	**96.18**	**93.13**	**95.175**

model is trained to detect and segment citrus fruits with high precision, capturing fine-grained details despite challenges like occlusions and complex textures. The second module evaluates the framework on full-tree images, leveraging the dual-image strategy and image-slicing techniques to address challenges such as dense foliage, overlapping fruits, and varying environmental conditions.

The results highlight the framework's robustness and adaptability across both protocols, showcasing its ability to perform effectively under diverse scenarios. The following subsection, Model Training Module Results, provides a detailed analysis of the framework's performance on close-up images.

3.1. Model training module results

At the completion of 2,000 iterations, the detection training accuracy was determined to be 98.8%, 99.0%, and 99.3% for stages 1, 2, and 3, respectively. The segmentation training accuracy was 96.7%. The total loss was 0.75.

A comparative analysis of various backbone models was then performed to assess the performance and robustness of the MViTv2 model. This analysis is essential for determining which backbone provides the highest accuracy and efficiency for object detection and segmentation tasks. Table 2 presents the Average Precision (AP) metrics on a testing dataset for different backbone models used in object detection and segmentation tasks. The backbones evaluated are Swin_L, ViTDet_L, and MViTv2_L, with the metrics divided into three categories: mAP, AP50, and AP75, each further split into bounding box (Bbox) and mask results.

From the table, MViTv2_L demonstrates superior performance across all metrics compared to the other backbones. For instance, it achieves the highest mAP scores for both bounding box (72.97%) and mask (84.40%) categories. Similarly, it records the highest values for AP50 with 98.60% for bounding box and 96.18% for mask, as well as for AP75 with 93.13% for bounding box and 95.175% for mask. In contrast, Swin_L performs lowest in all categories, indicating that MViTv2_L is the most effective backbone among those tested for object detection and segmentation tasks in this dataset, consolidating the choice for the detection and segmentation model.

The Figs. 6 illustrate the difference in detection for the three backbones. The images provided compare actual instances of fruit detection with predictions made using three different backbone models: MViTv2, Swin, and ViTDet. Each set of images includes an actual instance on the left and a predicted instance on the right.

In the first pair of images (Fig. 6a), derived from the Swin backbone, the predicted instance identifies multiple objects as fruits with varying confidence levels, illustrating the model's ability to detect multiple instances. However, some detections have lower confidence percentages, suggesting potential inaccuracies and false positives.

The second pair of images (Fig. 6b), related to the ViTDet backbone, the actual instance again shows a fruit among the leaves. The predicted instance identifies multiple fruits with bounding boxes and masks of different colors. This indicates that the ViTDet model can detect multiple objects, but some detections are less confident and may not be entirely accurate, similar to the Swin backbone.

In the third pair of images (Fig. 6c), generated using the MViTv2 backbone, the actual instance shows a single fruit among the leaves. The predicted instance correctly identifies and highlights the fruit with a bounding box and a mask, indicating high confidence (100%). This demonstrates MViTv2's strong capability in accurately detecting the fruit. Moreover, the MViTv2 model's strength is shown by the ability to detect and segment fruits not annotated in the testing dataset.

3.2. Segmentation and counting module

The proposed model's performance was compared to the manual counting of citrus fruits within visible images. The accuracy of the citrus counting was measured using the coefficient of determination (R2), the root mean squared error (RMSE), the relative RMSE (rRMSE), and the bias:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4}$$

$$\mathrm{rRMSE} = \left(\frac{\mathrm{RMSE}}{\bar{y}}\right) \times 100 \tag{5}$$

$$\mathrm{Bias} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i) \tag{6}$$

where n is the number of observations, y_i is the actual value, $\hat{y}_i$ is the predicted value, and $\bar{y}$ is the mean of the actual values.

To assess this module thoroughly, the model's detection performance on complete tree images was first examined with and without the slicing technique. The first step is applying the detection model trained in the previous module to the full tree images without slicing them. Fig. 7 presents the accuracy metrics' results.

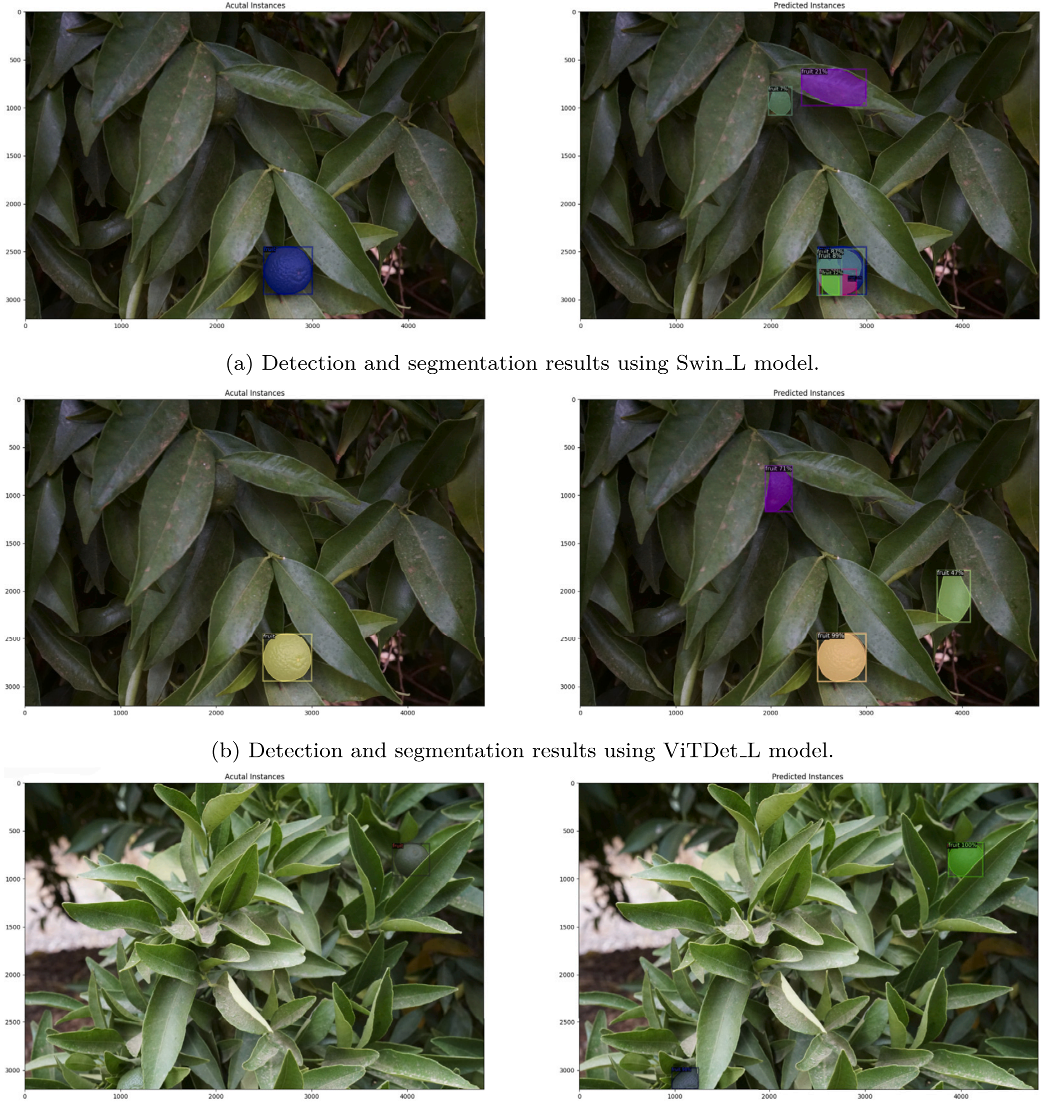
Based on the results shown in the figure, there is a low correlation between the manual count and the predicted count when applying the model to full tree images (R2 = 0.51). Moreover, the Bias is significantly high, with a value of -46.58, explaining the high underestimation of the citrus count when using the model. This proves the low accuracy of detection and segmentation of the model when used on full tree images.

The slicing strategy proposed in the module is then tested. For the first trial, the number of slices is set to 4 tiles per image. Fig. 8 illustrates the linear regression between the proposed model and the manual count tested on the 48 full trees images. The results showed a high correlation between the proposed framework's counting and the manual image-based counting. The model trained in the first module has a higher coefficient of determination and lower RMSE and rRMSE (R2 = 0.80, RMSE = 12.24, rRMSE = 15.62%), indicating that the model was closer to the visual observation. In addition, the bias value of -6.75 shows a slight underestimation of the number of citrus fruits compared to the visual assessment.

In order to gain a better understanding of the model's performance, the metrics for each variety were analyzed using the same number of slices (4S), which are illustrated in Fig. 9. Significant variations were observed in all metrics for each variety, with R-squared ranging from 0.58 to 0.81. Additionally, differences in Bias were noted, specifically an underestimation of count in both Nules grafted on Volka and Sidi Aissa grafted on Volka, as opposed to an overestimated count in Orogrande grafted on sour orange.

In this study on how different numbers of slices affect the module's performance, tests with various slicing values were conducted. Table 3 summarizes the impact of the number of slices on the framework performance, where NS presents the N number of slices per image, NsV represents Nules grafted on Volka, SIsV for Sidi Aissa grafted on Volka, and OsB for Orogrande grafted on sour orange.

The last step in this module involves Joining. During this step, the process of splitting the image is reversed to recombine the tiles into a single image complete with drawn segmentation, along with a CSV file containing the final count. Fig. 11 displays an example of the resulting combined image from the processed tiles presented in Fig. 10.



(a) Detection and segmentation results using Swin_L model.

(b) Detection and segmentation results using ViTDet_L model.

(c) Detection and segmentation results using MViTv2_L model.

Fig. 6. Results of Model Training Module using Cascade Mask R-CNN with **(a)** Swin backbone, **(b)** ViTDet backbone, and **(c)** MViTv2 backbone.

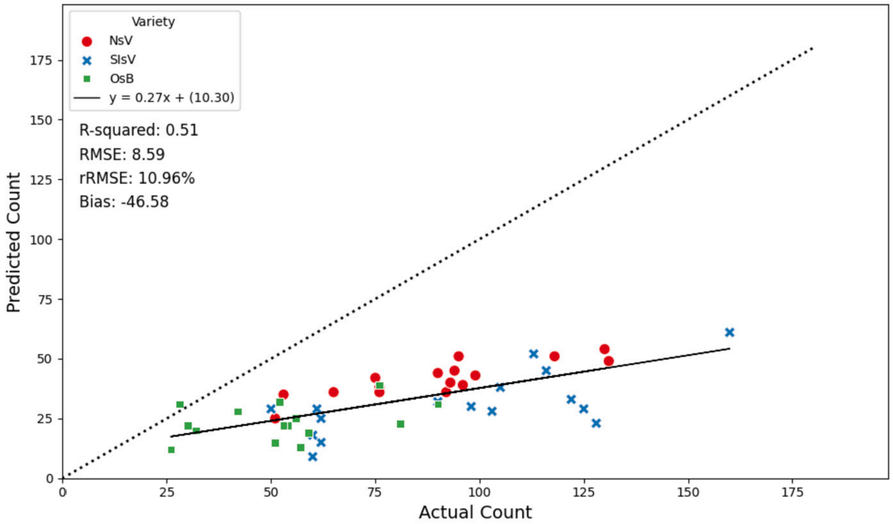


Fig. 7. Comparison of the number of citrus fruits visually counted on the full tree image with the number of fruits detected by the proposed model for all varieties: red dots represent Nules grafted on Volka, blue crosses represent Sidi Aissa grafted on Volka, and green squares for Orogrande grafted on sour orange.

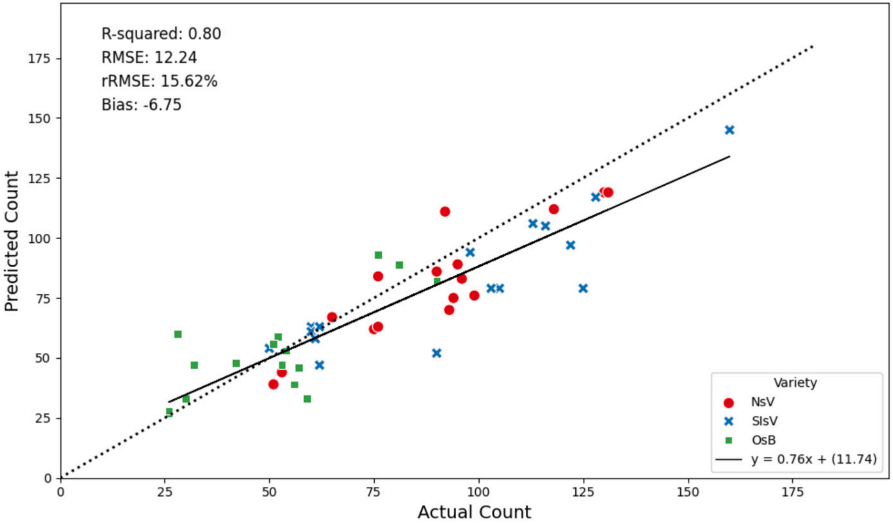


Fig. 8. Comparison of the total number of citrus fruits visually counted on the image with the number of fruits detected by the proposed slicing technique for all varieties: red dots represent Nules grafted on Volka, blue cross for Sidi Aissa grafted on Volka, and green squares for Orogrande grafted on sour orange.

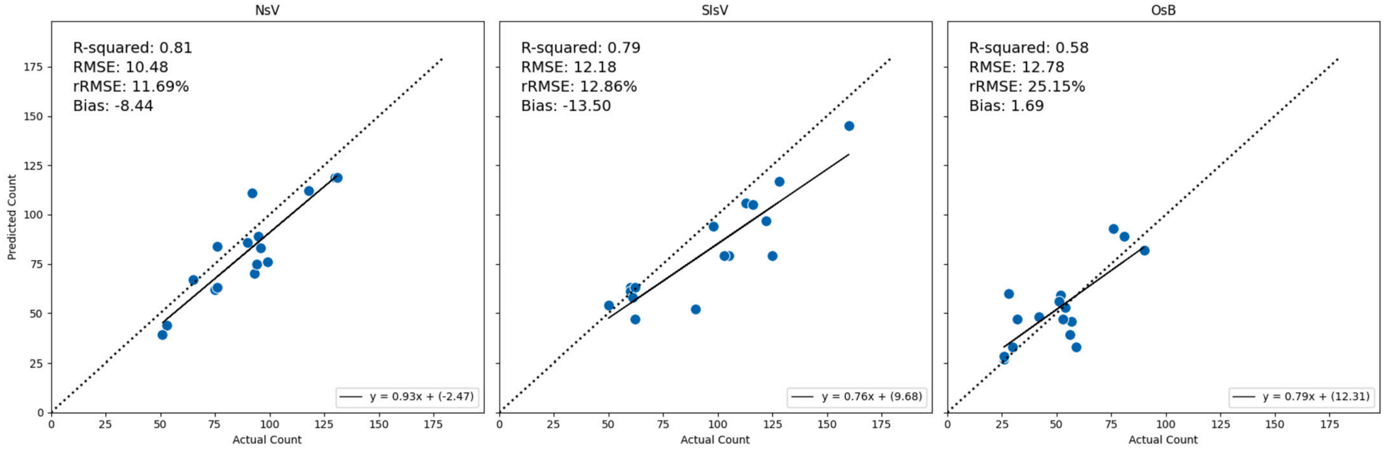


Fig. 9. Comparison of visually counted citrus fruits with detected fruits for each variety: NsV represents Nules grafted on Volka, SIsV for Sidi Aissa grafted on Volka, and OsB for Orogrande grafted on sour orange.

Table 3
Comparison of manually counted citrus per image and the number estimated by the proposed framework for three varieties with different numbers of slices.

	Metric	4S	6S	9S	12S	16S
NsV	R^2	0.81	0.81	0.92	0.93	0.87
	RMSE	10.48	11.33	6.41	6.27	8.25
	rRMSE	11.69%	12.64%	7.15%	7.00%	9.20%
	Bias	-8.44	-6.06	-14.25	-12.38	-13.38
SIsV	R^2	0.79	0.76	0.83	0.81	0.82
	RMSE	12.18	13.29	12.47	13.58	13.50
	rRMSE	12.86%	14.04%	13.17%	14.34%	14.26%
	Bias	-13.50	-13.19	-18.00	-16.00	-16.38
OsB	R^2	0.58	0.63	0.81	0.80	0.74
	RMSE	12.78	12.64	7.60	8.21	8.51
	rRMSE	25.15%	24.88%	14.96%	16.15%	16.75%
	Bias	1.69	2.00	-11.12	-10.25	-12.12

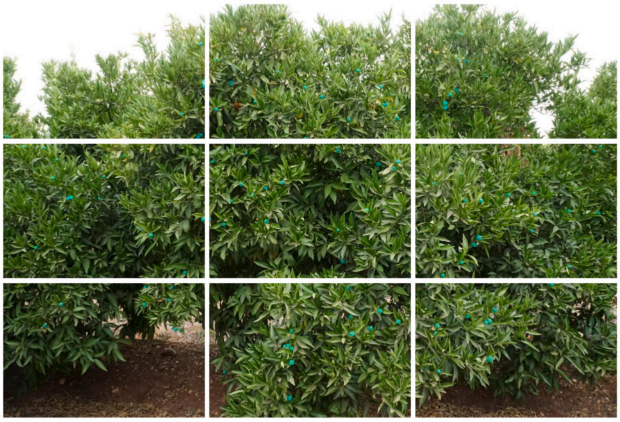


Fig. 10. Example of 9 tiles processed before joining.

4. Discussion

This study proposed a novel framework integrating MViTv2 and Cascade Mask R-CNN to enhance citrus fruit detection and segmentation in dense orchard environments. The research made several key assumptions: The dataset, comprising close-up and full-tree images of three citrus varieties, was assumed to represent real-world orchard conditions sufficiently. This included light variations, dense foliage, and occlusions, although extreme scenarios such as harsh weather or significant spatial differences between orchards were not comprehensively

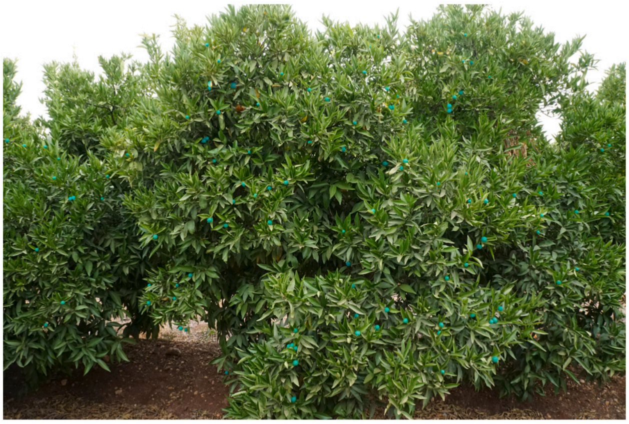


Fig. 11. Example of a resulting composite image after joining tiles.

covered. Additionally, the dual-image strategy assumed that features learned from close-up images could generalize effectively to full-tree images, enabling robust performance during testing. These assumptions provided a practical foundation for the study but highlighted the importance of expanding datasets and testing conditions for future work.

Several lessons were learned throughout the development and evaluation process. Integrating MViTv2 improved the framework's ability to capture fine-grained details, enabling accurate detection of small fruits obscured by foliage. The image-slicing technique further enhanced segmentation by focusing on localized image regions, significantly improving precision in challenging conditions. These innovations ensure that the framework can effectively adapt to the variability of citrus growth, environmental conditions, and varietal differences.

An important aspect of this framework is its design for practical, easy-to-use applications, such as deployment on smartphones or devices with simple setups, making it accessible for orchard managers, farmers, and citrus producers. By enabling early and accurate detection of citrus fruits using affordable and portable devices, this framework addresses a critical need in precision agriculture for scalable and cost-effective solutions. Using a dual-image strategy and image-slicing ensures the model is lightweight enough to be integrated into applications suitable for on-site use, making it a practical tool for real-time decision-making, including yield forecasting, without requiring expensive or complex equipment.

Building upon the assumptions and lessons learned, a detailed analysis of the framework's performance, along with its strengths and limitations, provides deeper insights into its applicability and areas for improvement. The following subsections comprehensively evaluate the framework and discuss the challenges and opportunities for further development.

4.1. Framework analysis

Throughout this paper, an AI-based framework was proposed that helps detect and segment small green citrus fruits in dense foliage at a very early stage. In the framework's first module, different backbones were compared to understand their influence on the performance of the Cascade Mask R-CNN model. As presented in Table 2, MViTv2_L achieved the highest performance compared to Swin_L and ViTDet_L across all metrics. This outcome points to the unique architectural design and training strategy as the key factors in the model's performance.

The chosen model demonstrated high performance with exceptional AP values during the evaluation. Nevertheless, a discrepancy between the actual and predicted counts was observed when the model on full tree images was tested. The correlation between the estimated and actual count was relatively poor, as presented in Fig. 7. Thus, the proposed slicing technique significantly improves the detection of green citrus fruits in full tree images (Fig. 8).

Furthermore, the variations in R2 and other evaluation metrics among the three varieties can be attributed to their size and growth stage factors. Notably, the Nules grafted on the Volka variety exhibits the highest R2 compared to the other varieties. This can be attributed to the fact that the Nules grafted on Volka variety entered the flowering stage earlier, on March 1st, 2022, while Sidi Aissa grafted on Volka, began flowering 15 days later, and Orogrande grafted on sour orange, started flowering on April 1st. These differences in flowering dates significantly impacted the citrus fruit size, resulting in noticeable variation in the detection, as shown in Fig. 9, making the model growth stage sensitive.

After assessing the impact of different slicing numbers on the module's performance (Table 3), it was observed that using 9 slices improved the detection and counting of citrus for both Sidi Aissa grafted on Volka and Orogrande grafted on sour orange, with R2 values of 0.83 and 0.81, respectively. However, for Nules grafted on Volka, the findings showed that 12 slices per image slightly outperformed 9 slices. Upon visual inspection, the detections were inaccurate despite the higher estimated number of citrus using 12 slices. Increased slicing led to overlapping, resulting in repeated counting, as illustrated in the Fig. 12.

These observations revealed that while 12 slices exhibited higher metric values compared to 9 slices, the latter demonstrated superior accuracy in detecting citrus fruits. In contrast, 12 slices resulted in significantly more false positives. Fig. 13 provides an example of the difference in detection between 9 slices and 12 slices, where two citrus fruits were accurately detected using the 9 slices strategy, resulting in the high performance of 9 slices techniques compared to the other slices numbers.

Overall, the 9 slices performed significantly better in accurately detecting and counting all varieties, despite differences in citrus' growth stages and sizes. This improvement demonstrates the effectiveness of the slicing technique in enhancing the model's counting accuracy.

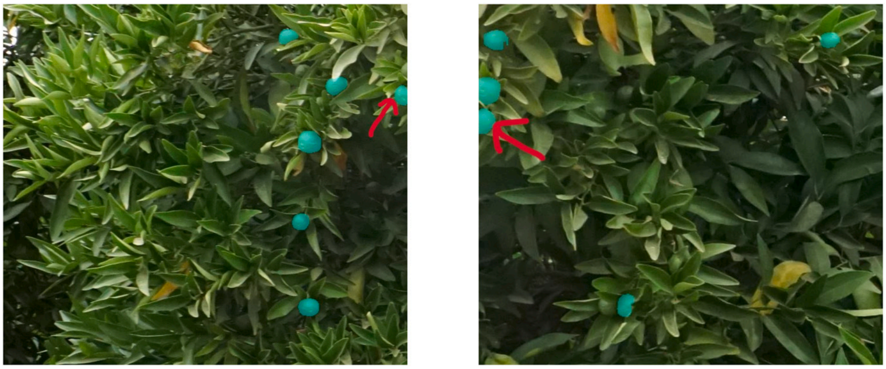


Fig. 12. Example of two tiles of image using the 12S. The red arrows point to the citrus detected repeatedly in both tiles.

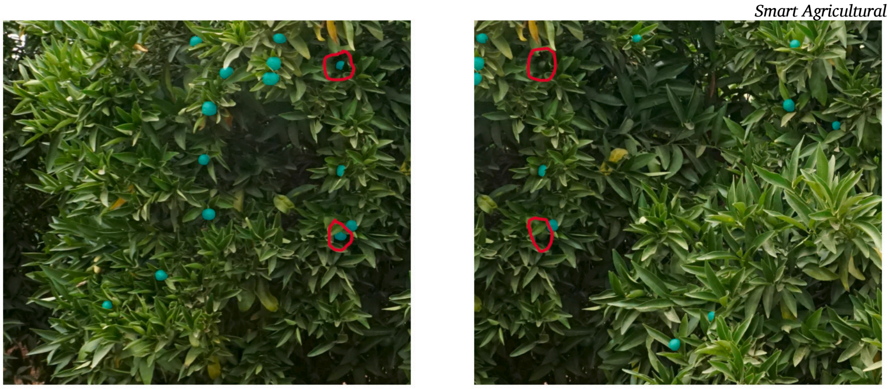


Fig. 13. Example of two tiles of the image with drawn citrus detection: (a) presents the detection in 9 slices, (b) presents the detection in 12 slices. The red circles show the difference in detection.

The literature shows diverse research on green citrus detection and counting in the expansive agricultural research and technology realm. He et al. [15] proposed a green fruit detection method named deep bounding box regression forest (DBBRF) for detecting green citrus fruits in natural environments and achieved a mAP of 87.60%, while Zheng et al. [41] proposed a method of green citrus detection using a deep convolutional neural network, combining the strength of multi-scale convolutional neural network and YOLO, and achieved a mAP of 91.55%. Lyu et al. [24] proposed a YOLOv5-CS model combined with an AI edge system for detecting and counting green citrus fruits. Their model achieved an mAP of 91.55%, accuracy of 86%, and recall of 91%. In addition, Lu et al. [23] presented a lightweight green citrus fruit detection model suitable for edge smart devices, achieving a mAP of 93.6%. These results exhibit the strength and high performance of the detection models. However, the images used for testing are very similar to the trained data, where the models detect and count citrus fruits in a close view of the tree, where citrus can be visible and easily detected. In addition, the models used are mainly based on object detection, unlike this study's framework, in which instance segmentation was incorporated.

In the proposed framework, the tested images differ significantly from the training data, posing a considerable challenge for detection and counting, even to the human eye. The proposed framework aims to count the citrus fruits on the tree accurately and includes instance segmentation, which provides additional details such as the pixel size and precise location of each detected citrus fruit in the image. These noteworthy results represent an innovative approach to early-stage detection, segmentation, and counting of citrus fruits.

4.2. Limitations and future perspectives

While the framework proposed in this paper demonstrates significant advancements in citrus fruit detection and segmentation, it is not without limitations. One of the primary challenges lies in accurately detecting and counting citrus fruits of varying sizes, especially when the size difference is substantial. Smaller fruits can be particularly difficult to detect, leading to undercounting, while larger fruits may be counted multiple times if they overlap with other elements in the image. Additionally, shadows cast by foliage and branches create regions of low visibility, obscuring fruits and causing false negatives or false positives.

Another notable limitation is the handling of overlapped slices and the issue of sliced citrus fruits leading to incomplete information or repeated counting. When images are divided into slices to facilitate detection, there is a risk of counting the same fruit multiple times if it appears in more than one slice. This overlap can result in an overestimation of fruit counts. Conversely, if a fruit is partially visible in multiple slices but not fully captured in any single slice, it might not be counted, leading to underestimation. These challenges necessitate the development of more sophisticated algorithms capable of recognizing and reconciling these overlaps to ensure accurate counting.

Furthermore, increasing the number of slices enhances the accuracy of detection but also significantly increases the processing time, making the system less efficient. This trade-off between accuracy and processing speed is a critical limitation, especially for real-time applications. Future solutions could involve optimizing the slicing strategy to balance accuracy and efficiency, possibly through adaptive slicing techniques that vary based on image complexity.

Future research should focus on several key areas to address these limitations. First, enhancing the model's ability to differentiate between individual fruits and their segments by integrating more advanced image-slicing techniques can mitigate issues related to overlapped slices. In addition, future work will focus on augmenting the dataset with more diverse environmental conditions, such as varying light intensities and weather scenarios. Finally, exploring the potential of other machine learning architectures may also offer new insights and improvements in fruit detection and counting accuracy. These advancements will pave the way for more reliable and accurate fruit detection systems in real-world agricultural applications.

5. Conclusion

In modern agricultural research, instance segmentation is a vital tool for enhancing the accuracy and precision of crop analysis. This is particularly important for citrus fruits, where detailed information about each fruit's size, location, and segmentation is crucial for assessing health, growth, and yield estimations.

The Cascade Mask R-CNN algorithm, paired with the MViTv2.L backbone, has proven highly effective. It excels in detecting and segmenting individual citrus fruits, providing precise masks invaluable for detailed phenotyping analyses and identifying potential anomalies or diseases. The backbone networks in Cascade Mask R-CNN are essential for feature extraction, which is critical for the model's overall performance. The choice of the MViTv2.L backbone enhances the quality of features extracted from images, improving the accuracy and speed of the region proposal and segmentation processes. Given the unique characteristics of citrus fruits, such as their size, shape, and varying light conditions, selecting an appropriate backbone is vital for optimizing segmentation accuracy.

However, the intricate nature of citrus orchards presents challenges for the Cascade Mask R-CNN model. The varying sizes, shapes, and overlapping of fruits, along with shadows from foliage, can impede accurate segmentation. Additionally, slicing the images to improve detection can lead to incomplete information or repeated counts, necessitating further model refinement and preprocessing techniques to enhance detection robustness.

This research introduced a novel framework using the MViTv2.L backbone and a slicing strategy. This approach significantly improved detection and segmentation accuracy by enabling the model to handle dense foliage and varying fruit orientations better. The slicing strategy divides images into smaller sections, reducing the complexity of each segment and allowing for more precise identification and counting of fruits. This method effectively mitigates the challenges posed by overlapping fruits and shadows, resulting in higher accuracy and more reliable segmentation.

In conclusion, integrating Cascade Mask R-CNN with the MViTv2.L backbone and a strategic slicing technique has shown promising results in citrus fruit detection and segmentation. Despite the challenges, these innovations represent a significant step forward in agricultural technology, providing a robust tool for precise crop analysis. Future work will further refine these techniques and explore additional improvements to enhance model performance and applicability in various agricultural contexts.

CRediT authorship contribution statement

Manal El Akrouchi: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Formal analysis, Conceptualization. **Manal Mhada:** Writing – review & editing, Validation, Resources, Project administration, Data curation. **Mohamed Bayad:** Writing – review & editing, Resources, Data curation. **Malcolm J. Hawkesford:** Writing – review & editing, Supervision, Investigation. **Bruno Gérard:** Writing – review & editing, Validation, Supervision, Project administration, Investigation, Funding acquisition.

Funding

This research was funded by OCP Morocco and the University Mohammed VI Polytechnic, project Rf.: FP04.

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.

Data availability

Data will be made available on request.

References

[1] F.C. Akyon, S.O. Altinuc, A. Temizel, Slicing aided hyper inference and fine-tuning for small object detection, in: 2022 IEEE International Conference on Image Processing (ICIP), IEEE, 2022, pp. 966–970.

[2] A. Azizi, Z. Zhang, W. Hua, M. Li, C. Igathinathane, L. Yang, Y. Ampatzidis, M. Ghasemi-Varnamkhasti, M. Zhang, H. Li, et al., Image processing and artificial intelligence for apple detection and localization: a comprehensive review, Comput. Sci. Rev. 54 (2024) 100690.

[3] J. Brown, The Economics of Citrus Growing, Springer, 2019.

[4] Z. Cai, N. Vasconcelos, Cascade r-cnn: delving into high quality object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6154–6162.

[5] G. Chen, C. Hua, Z. Zhang, Recognition of citrus on trees in different growth periods, in: 2023 42nd Chinese Control Conference (CCC), IEEE, 2023, pp. 7321–7326.

[6] Y. Cheng, L. Li, Y. Xu, X. Li, Z. Yang, W. Wang, Y. Yang, Segment and track anything, arXiv preprint, arXiv:2305.06558, 2023.

[7] D. Choi, W.S. Lee, R. Ehsani, J.K. Schueller, F. Roka, Machine vision system for early yield estimation of citrus in a site-specific manner, in: 2015 ASABE Annual International Meeting, American Society of Agricultural and Biological Engineers, 2015, p. 1.

[8] U.O. Dorj, M. Lee, S.s. Yun, An yield estimation in citrus orchards via fruit detection and counting using image processing, Comput. Electron. Agric. 140 (2017) 103–112.

[9] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, Multiscale vision transformers, arXiv preprint, arXiv:2104.11227, 2021.

[10] F. Garcia-Ruiz, S. Sankaran, J.M. Maja, Spectral imaging and analysis for crop management, Remote Sens. Appl. Agric. 2 (2015) 85–103.

[11] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.Y. Lin, E.D. Cubuk, Q.V. Le, B. Zoph, Simple copy-paste is a strong data augmentation method for instance segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2918–2928.

[12] K.A. Hashmi, A. Pagani, M. Liwicki, D. Stricker, M.Z. Afzal, Cascade network with deformable composite backbone for formula detection in scanned document images, Appl. Sci. 11 (2021) 7610.

[13] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.

[14] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, IEEE Trans. Pattern Anal. Mach. Intell. (2020), https://doi.org/10.1109/tpami.2018.2844175.

[15] Z. He, J. Xiong, S. Chen, Z. Li, S. Chen, Z. Zhong, Z. Yang, A method of green citrus detection based on a deep bounding box regression forest, Biosyst. Eng. 193 (2020) 206–215.

[16] S. Jones, M. Roberts, Ai techniques in agriculture: a review, J. Agric. Syst. 22 (2018) 105–119.

[17] A. Kamilaris, F.X. Prenafeta-Boldú, Deep learning in agriculture: a survey, Comput. Electron. Agric. 147 (2018) 70–90.

[18] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A.C. Berg, W.Y. Lo, et al., Segment anything, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 4015–4026.

[19] X. Li, Q. Wang, A review on the methods of precision agriculture, Agric. Sci. 8 (2017) 666–675.

[20] Y. Li, C.Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, C. Feichtenhofer, Mvitv2: improved multiscale vision transformers for classification and detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 4804–4814.

[21] Z. Li, F. Zhou, Cascade r-cnn: high quality object detection and instance segmentation, IEEE Trans. Image Process. 29 (2020) 930–944.

[22] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv:1711.05101, 2019.

[23] J. Lu, P. Chen, C. Yu, Y. Lan, L. Yu, R. Yang, H. Niu, H. Chang, J. Yuan, L. Wang, Lightweight green citrus fruit detection method for practical environmental applications, Comput. Electron. Agric. 215 (2023) 108205, https://doi.org/10.1016/j.compag.2023.108205.

[24] S. Lyu, R. Li, Y. Zhao, Z. Li, R. Fan, S. Liu, Green citrus detection and counting in orchards based on yolov5-cs and ai edge system, Sensors 22 (2022) 576.

[25] M. Mhamed, Z. Zhang, W. Hua, L. Yang, M. Huang, X. Li, T. Bai, H. Li, M. Zhang, Apple varieties and growth prediction with time series classification based on deep learning to impact the harvesting decisions, Comput. Ind. 164 (2025) 104191.

[26] M. Mhamed, Z. Zhang, J. Yu, Y. Li, M. Zhang, Advances in apple's automated orchard equipment: a comprehensive research, Comput. Electron. Agric. 221 (2024) 108926.

[27] H.Y. Oh, M.S. Khan, S.B. Jeon, M.H. Jeong, Automated detection of greenhouse structures using cascade mask r-cnn, Appl. Sci. 12 (2022) 5553.

[28] L. Perez, J. Wang, The effectiveness of data augmentation in image classification using deep learning, arXiv:1712.04621, 2017.

[29] X. Qin, R. Huang, B. Hua, Research and implementation of yield recognition of citrus reticulata based on target detection, in: Journal of Physics: Conference Series, IOP Publishing, 2021, p. 012128.

[30] Z. Rui, Z. Zhang, M. Zhang, A. Azizi, C. Igathinathane, H. Cen, S. Vougioukas, H. Li, J. Zhang, Y. Jiang, et al., High-throughput proximal ground crop phenotyping systems–a comprehensive review, Comput. Electron. Agric. 224 (2024) 109108.

[31] K.P. Seng, et al., Deep learning for image-based citrus fruit detection, J. Imag. 6 (2020) 10.

[32] D. Su, Y. Qiao, Y. Jiang, J. Valente, Z. Zhang, D. He, Ai, Sensors and Robotics in Plant Phenotyping and Precision Agriculture, Volume ii, 2023.

[33] W. Sun, Z. Liu, Y. Zhang, Y. Zhong, N. Barnes, An alternative to wsss? An empirical study of the segment anything model (sam) on weakly-supervised semantic segmentation problems, arXiv preprint, arXiv:2305.01586, 2023.

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S.M. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[35] Y. Tianjing, M. Mhamed, Developments in automated harvesting equipment for the apple in the orchard, Smart Agric. Technol. (2024) 100491.

[36] X. Wu, L.R. Khot, S. Sankaran, Object detection and tracking in agricultural fields: challenges and opportunities, IEEE IT Prof. 22 (2020) 34–42.

[37] Y. Wu, A. Kirillov, F. Massa, W.Y. Lo, R. Girshick, Detectron2, https://github.com/facebookresearch/detectron2, 2019.

[38] Z.F. Yang, F. Xiao, Y. Ding, Y. Ding, M.J. Paul, Z. Liu, Leaf to panicle ratio (lpr): a new physiological trait for rice plant architecture based on deep learning, https://doi.org/10.21203/rs.3.rs-25185/v1, 2020.

[39] Z. Zhang, P. Flores, A. Friskop, Z. Liu, C. Igathinathane, X. Han, H. Kim, N. Jahan, J. Mathew, S. Shreya, Enhancing wheat disease diagnosis in a greenhouse using image deep features and parallel feature fusion, Front. Plant Sci. 13 (2022) 834447.

[40] Z. Zhang, Y. Lu, R. Lu, Development and evaluation of an apple infield grading and sorting system, Postharvest Biol. Technol. 180 (2021) 111588.

[41] Z. Zheng, J. Xiong, H. Lin, Y. Han, B. Sun, Z. Xie, Z. Yang, C. Wang, A method of green citrus detection in natural environments using a deep convolutional neural network, Front. Plant Sci. 12 (2021) 705737.

[42] D. Zhou, J. Fang, X. Song, C. Guan, J. Yin, Y. Dai, R. Yang, Iou loss for 2d/3d object detection, arXiv:1908.03851, 2019.