Computers and Electronics in Agriculture 248 (2026) 111833

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Large-scale orange fruit dataset for localization, classification and ripening assessment under varying environments

Alessandro Carella [a,1,*], Baptiste Paul Ernest Lucas [b,1], Safouane El Ghazouali [b,*,1], Pedro Tomas Bulacio Fischer [a], Roberto Massenti [a], Francesca Venturini [b,c], Umberto Michelucci [b,d], Riccardo Lo Bianco [a]

[a] Department of Agricultural, Food and Forest Sciences (SAAF), University of Palermo, 90128 Palermo, Italy
[b] TOELT LLC Machine Learning Research and Development, Birchlenstrasse 25, 8600 Duebendorf, Switzerland
[c] ZHAW Zurich University of Applied Sciences, School of Engineering, Institute of Applied Mathematics and Physics, 8401 Winterthur, Switzerland
[d] HSLU Lucerne University of Applied Sciences and Arts, Computer Science Department, 6343 Risch-Rotkreuz, Switzerland

ARTICLE INFO

Dataset link: Oranges in the field (Original data)

Keywords:
Computer vision
Deep learning
YOLO
CLIP
Object detection
Citrus
Fruit detection

ABSTRACT

The presented dataset is a large-scale on-field orange image collection designed for fruit detection, classification, and ripening assessment in real-world orchard environments. It includes 5025 images captured under diverse weather, lighting conditions, and ripening stages. Images were collected using different smartphone cameras and preprocessed through a custom cropping algorithm to optimize annotation efficiency. The dataset was labeled using a semi-automated approach, combining YOLO-based pre-annotations refined manually in Roboflow. Annotation quality was further improved through a CLIP-based verification process to filter incorrect labels. The dataset is released with both YOLO and COCO annotations, enabling compatibility with multiple object detection frameworks. Additionally, a benchmark evaluation was conducted using state-of-the-art models, including YOLO (v5, v8, v10, v11) and RT-DETR, assessed via standard precision, recall, and F1-score metrics. Results showed that recent YOLO models (YOLOv10 and YOLOv11) achieved high detection performance, with mAP@0.5 values close to 0.892, and consistently outperform RT-DETR baselines (mAP@0.5 = 0.851) in terms of precision and inference speed. In this context, the structured design and high environmental diversity of the proposed dataset make it a valuable resource for developing and evaluating computer vision solutions in precision agriculture, including fruit ripening assessment, yield estimation, and automated harvesting.

1. Introduction

The integration of computer vision into agriculture has become crucial for advancing precision horticulture and addressing the growing need for automation in fruit production systems (Kamilaris and Prenafeta-Boldú, 2018; Vasconez et al., 2020; Wang et al., 2022). Accurate detection and localization of fruits in orchards are fundamental for tasks such as yield estimation, robotic harvesting, monitoring of plant health, and fruit quality and ripening assessment (Mirhaji et al., 2021; Massah et al., 2021; Zhang et al., 2023). However, collecting datasets that support robust model development remains a substantial challenge, especially in open-field citrus environments. Despite the increasing availability of fruit detection datasets for crops such as apple (Bortolotti et al., 2024; Wu et al., 2024), pear (Bonora et al., 2021), mango (Shi et al., 2020), pomegranate (J. Zhao et al., 2024), and chestnut (Arakawa et al., 2024), public datasets specifically designed for citrus fruits, especially for orange detection under diverse field conditions, are still limited. While some existing works have addressed variable lighting (Mirhaji et al., 2021) and occlusion scenarios (Lin et al., 2024; Xiao et al., 2024), they often focus on specific cultivars or use earlier computer vision models without providing reusable open datasets. In this context, this work contributes by releasing a large-scale, publicly available orange fruit dataset collected under highly heterogeneous conditions, covering various weather situations, times of day, and fruit ripening stages. Moreover, it introduces a prompt-based CLIP (Contrastive Language-Image Pre-Training) (Radford et al., 2021) verification pipeline to ensure annotation quality and maximize generalizability across diverse environments. Fruit detection in citrus orchards is particularly complex due to occlusions by leaves

* Corresponding authors.
E-mail addresses: alessandro.carella@unipa.it (A. Carella), safouane.elghazouali@toelt.ai (S. El Ghazouali).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.compag.2026.111833
Received 23 October 2025; Received in revised form 14 April 2026; Accepted 23 April 2026
Available online 28 April 2026

and branches, wide variations in fruit color and size across ripening stages, and the presence of visually confusing backgrounds such as bark, weeds, or netting (Oviedo Espinosa et al., 2024). These factors reduce the efficacy of deep learning models trained on narrow or homogeneous datasets.

A curated dataset of orange fruits, captured under diverse and uncontrolled field conditions over a three-month period, was developed to address these limitations. These include combinations of three times of day (morning, afternoon, evening) and three weather conditions (sunny, cloudy, rainy), resulting in a highly varied image corpus. A key novelty of this dataset lies in its emphasis on environmental diversity and label precision. A hybrid semi-automatic annotation pipeline, combining YOLO (You Only Look Once) fine-tuning and manual correction, was used to ensure high labeling consistency across all images (Lin et al., 2024). Furthermore, a zero-shot visual classification step using CLIP (Radford et al., 2021) was introduced to identify and discard incorrectly labeled or fruitless bounding boxes — an uncommon practice in current agricultural datasets (Nawaz et al., 2024). The dataset is accompanied by a detailed description of the acquisition and annotation pipeline, as well as a comprehensive benchmark of fruit detection models, including YOLOv5 through YOLOv11 and RT-DETR, trained on the dataset. To the best of our knowledge, this is the first citrus fruit detection dataset that combines rich weather annotation, rigorous filtering with CLIP, and multi-device acquisition protocols. The ability to accurately detect and classify oranges in real field conditions makes this dataset a valuable resource for several essential agricultural operations. These include estimating fruit ripening to optimize harvest timing, predicting yield to improve supply chain planning, and developing autonomous or semi-autonomous harvesting systems. Fruit maturation is a complex physiological process that varies significantly not only across orchards but also within individual trees, due to differences in canopy exposure, fruit load, and microclimatic conditions (Carella et al., 2023). Climate variability, due to natural fluctuations or longer-term climate change, further amplifies spatial and temporal heterogeneity in fruit development. In citrus species such as sweet orange, fruit quality is closely linked to parameters such as size, juice content, soluble sugars (°Brix), acidity, and carotenoid coloration, all of which directly affect market value, postharvest performance, and consumer preference (Mossad et al., 2020; Massenti et al., 2016). Accurate, spatially resolved, and timely monitoring of fruit ripening is therefore essential to optimize harvest timing, reduce waste, and improve overall profitability. These operations are particularly challenging in open-field citrus orchards, where strong occlusions, overlapping canopies, variable lighting, and cultivar-specific traits often compromise detection performance (James et al., 2024). The dataset addresses these challenges by capturing highly diverse scenarios in terms of lighting, weather, and fruit appearance. As a result, it can also support downstream tasks such as orchard digitization, selective fruit thinning, and targeted pesticide application based on accurate fruit localization and ripening stage. Overall, the dataset contributes to the development of scalable AI-driven solutions for precision agriculture, with the potential to improve sustainability, reduce labor dependency, and enhance decision-making in citrus production systems.

To the best of our knowledge, the main novelties of this work can be summarized as follows: (i) the release of a large-scale, publicly available on-field orange dataset captured under highly heterogeneous real-world conditions, including diverse weather scenarios, illumination regimes, and fruit ripening stages; (ii) the adoption of a hybrid annotation pipeline that combines YOLO pre-annotation, manual refinement, and a CLIP verification step to improve label reliability and reduce annotation noise; (iii) the inclusion of rich contextual metadata and a comprehensive benchmark of both convolutional (YOLO) and transformer-based (RT-DETR) object detection models, enabling reproducible evaluation under challenging orchard conditions. Together, these contributions address key limitations of existing citrus datasets and provide a robust foundation for developing and benchmarking computer vision methods in precision horticulture. By providing this dataset, the present work aims to support reproducible research and further advances in fruit detection under real-world orchard conditions.

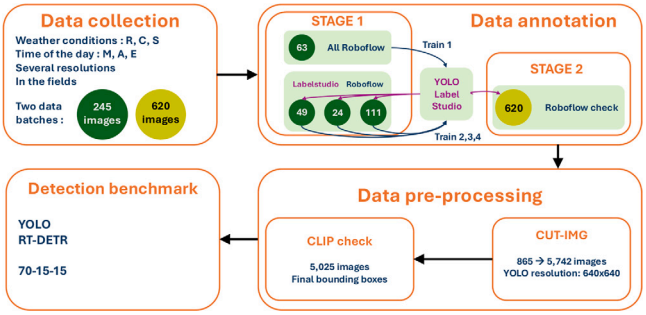

Fig. 1. Complete data pipeline developed for the creation of the orange fruit dataset. The pipeline is structured into four main stages: (1) Data collection under multiple weather and lighting conditions; (2) Semi-automated annotation using a YOLO-based pre-labeling system refined via Roboflow and Label Studio; (3) Preprocessing with the CUT-IMG algorithm and CLIP-based filtering to generate high-quality sub-images; and (4) A benchmark evaluation using multiple object detection models.

2. Materials and methods

The dataset was developed through a newly designed pipeline aimed at producing high-quality images and annotations. The pipeline, summarized in Fig. 1, consisted of four main steps:

- Data collection: this step involved large-scale image capturing under multiple environmental and lighting conditions to diversify the dataset, improving generalization and increasing detection performance. The images were acquired from multiple citrus orchards maintained by the University of Palermo, using several commercial smartphones to promote diversity in resolution and optical characteristics. The dataset included over 5000 sub-images derived from 865 original images collected in situ under nine defined weather–lighting conditions.
- Data annotation: this process consisted of two sub-stages. First, the labeling process began with training a YOLO-based labeling robot on an initial batch of annotated images using Label Studio (Heartex, San Francisco, CA, USA). The robot then pre-annotated the remaining images, which were manually refined in Roboflow (Dwyer et al., 2024) to ensure each bounding box contained exactly one fruit with minimal background.
- Pre-processing: again, this step consisted of two sub-processes. First, the CUT-IMG algorithm was developed to split the initial annotated images into square sub-images while preserving bounding box integrity, ensuring a YOLO-standardized format. To further ensure the quality of the resulting sub-images, a CLIP-based zero-shot classifier was used to detect those with incorrect or empty bounding boxes by evaluating their visual content against a set of 17 manually defined textual prompts.
- Detection benchmark: Finally, multiple object detection models were trained on the final dataset to assess data quality, accuracy, and real-time performance using standard detection metrics.

2.1. Data collection

Images were acquired at the University of Palermo experimental orchards (38°06′24″N 13°21′05″E), as well as from additional commercial orchards located in the southwestern area of Sicily, between Ribera and Burgio (AG). Image acquisition was intentionally designed to be unconstrained with respect to camera pose, viewing angle, and acquisition distance. No fixed protocol regarding camera height, distance from the tree, or shooting angle was imposed. This choice was motivated by the objective of maximizing real-world variability and reflecting realistic operating conditions for fruit detection

Fig. 2. Representative examples of image acquisition conditions during data collection. Panels (a) and (b) show full-tree views captured at different distances and under varying lighting conditions. Panels (c) and (d) show closer canopy-level views with different degrees of fruit ripening.

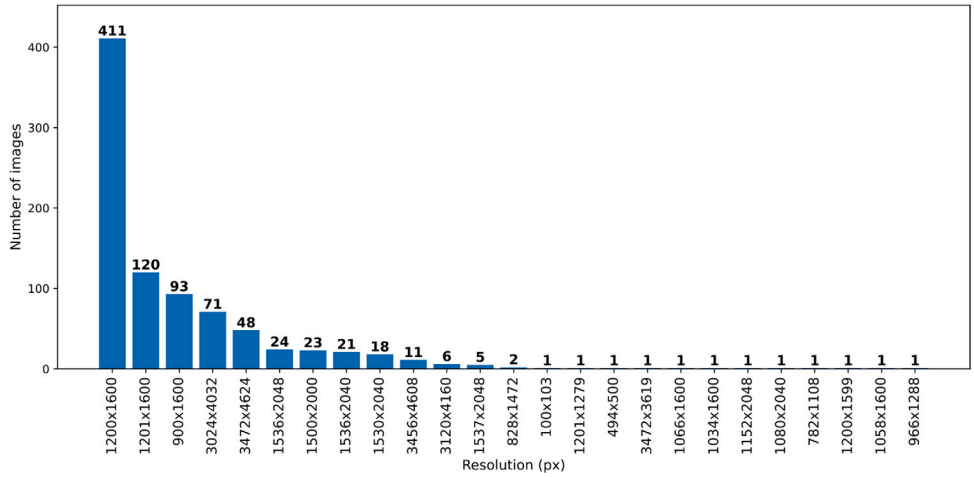

Fig. 3. Distribution of the original resolutions of the 865 images captured in the field. The dataset includes a broad range of resolutions resulting from the use of seven different smartphone models.

systems deployed in agricultural environments. Acquisition distances ranged from approximately 15 m, beyond which fruit visibility and annotation reliability significantly decreased (Fig. 2a–b), to close-range views (Fig. 2c–d). Both partial views of canopies and full-tree perspectives were collected, depending on the size and training system of the trees. Tree characteristics such as age, canopy volume, and vigor varied substantially, as images were collected from both experimental orchards and commercial production systems. Image acquisition was further conducted under a wide range of environmental and lighting conditions, intentionally capturing scenes at different times of day (morning, afternoon, evening, and night) and under diverse weather conditions (sunny, cloudy and rainy scenarios). This variability was explicitly introduced during the data collection stage and later formalized during dataset organization, as described in Section 2.5.

To ensure resolution and sensor diversity, seven different smartphone models were employed: Apple iPhone 13, iPhone 11, iPhone SE 2020, iPhone 8, Google Pixel 8, Samsung Galaxy A34, and Samsung Galaxy A13. These devices offer a wide range of camera capabilities, from single 12 MP sensors to 50+ MP multi-camera systems, with varying focal lengths, color processing profiles, and embedded image optimization pipelines (Table 1). A histogram of the resolutions of the 865 initial images is also available in Fig. 3.

This heterogeneity in hardware contributed to a dataset encompassing a broader range of sharpness levels, dynamic range behaviors, white balance calibrations, and compression artifacts, factors that are critical for training models robust to real-world variability. To capture seasonal and environmental variability, image acquisition was conducted at different times of day and under varying weather conditions, between October and December 2024. Trees from five distinct cultivars of sweet orange (*Citrus sinensis* (L.) Osbeck) were included in the dataset: Navelina (early ripening cv), Washington Navel and Tarocco (mid-season cv), and Sanguinello Moscato and Valencia (late ripening cv). This assortment ensured the inclusion of oranges at different developmental stages and colorations, contributing to the diversity of visual characteristics in the dataset.

Table 1
Main technical specifications of the seven smartphone models used to collect the dataset images. Each device's configuration is detailed, including the resolution and types of lenses (e.g., wide, ultra-wide, macro).

Device	Main camera specifications
iPhone 13	Dual 12 MP (Wide f/1.6, Ultra-wide f/2.4)
iPhone 11	Dual 12 MP (Wide f/1.8, Ultra-wide f/2.4)
iPhone SE 2020	12 MP (Wide f/1.8)
iPhone 8	12 MP (Wide f/1.8)
Google Pixel 8	50 MP (Wide f/1.68) + 12 MP (Ultra-wide f/2.2)
Samsung Galaxy A34	48 MP (Wide f/1.8) + 8 MP (Ultra-wide) + 5 MP (Macro)
Samsung Galaxy A13	50 MP (Wide f/1.8) + 5 MP (Ultra-wide) + 2 MP (Macro) + 2 MP (Depth)

2.2. Data annotation

The newly collected images were stored in specific folders in the Roboflow (Dwyer et al., 2024) annotation tool under the appropriate category (one of nine, each representing a different weather condition). All the images were labeled in two steps.

The first stage involved training and refining a labeling robot by fine-tuning a YOLO model on an initial batch of annotated images split in several sub-batches. This was done using Label Studio, with details provided in the YOLO-Label Studio section (Fig. 4).

The second stage consisted of passing all remaining images through the labeling robot, providing a first labeling version for each image. The pre-annotated images were then re-uploaded to Roboflow, where the labels were manually refined through individual inspection. A straightforward labeling strategy was adopted, assigning one fruit per bounding box while avoiding partial fruit cropping and unnecessary background. Only fruits that were clearly recognizable as oranges by visual inspection were annotated. To ensure annotation reliability and bounding-box accuracy, fruits that appeared excessively small in the image or whose identity was uncertain due to distance or severe blur were excluded. In practice, this corresponded to excluding fruits captured at distances greater than approximately 15 m, where the number of pixels per fruit was insufficient for confident identification. Partially occluded fruits were annotated only when the visible portion allowed a reliable estimation of fruit size and shape. As a general criterion, fruits with an occlusion level exceeding approximately 80% were not annotated. This strategy aimed to balance dataset realism with the need for consistent and meaningful bounding-box annotations.

YOLO-label studio. During the data annotation process, a small initial dataset of 245 images of orange trees was prepared to train a YOLO architecture to be used as a labeling robot. All images were manually annotated. A backend connection was established between a Python script and the Label Studio platform. The backend YOLO model was initially trained on a subset of 63 images to obtain a functional version of the annotation tool. This preliminary model was then used to pre-annotate the remaining 182 images in a recursive manner: a portion of the dataset was pre-annotated through Label Studio, the annotations were completed and refined in Roboflow, and the resulting dataset was used to re-train the YOLO model. This iterative process was repeated four times, progressively incorporating new batches of annotated images.

The YOLO model used for the labeling is YOLOv8n from the `Ultralytics` framework and fine-tuned on a single NVIDIA GPU with the following hyperparameters:

- Number of epochs: 50
- batch size: 32
- optimizer: AdamW
- momentum: 0.937
- learning rate: 0.01
- weight decay: 0.0005

In Label Studio, the confidence threshold slider was dynamically adjusted (usually between 0.25 and 0.5) to control the number of proposed bounding boxes, balancing automation speed and the need for manual correction.

Criteria for accepting or correcting pre-annotations during manual refinement in Roboflow were as follows: a proposed bounding box was accepted if it tightly enclosed exactly one complete fruit with minimal background and high visual alignment (implicitly high IoU with the expected position). Boxes with low confidence, misalignment, overlap with multiple fruits, or false positives (e.g., leaves or branches) were corrected by adjusting coordinates or deleted. Missed fruits were manually added as new bounding boxes following the same tight-fitting strategy. This ensured high annotation consistency across the entire dataset.

2.3. Data pre-processing

CUT-IMG algorithm. As illustrated in Fig. 5, the final dataset consisted of square YOLO-formatted images, i.e. (640,640). However, larger images were initially used and labeled in order to capture the highest possible number of fruits. This would have taken more time on sub-images as they showed partial redundancy, it therefore reduced the labeling time. To achieve this, a small algorithm (named CUT-IMG) was developed to split images into sub-images (available in the final code). The algorithm operates as follows: (1) if necessary, the original image is padded (with reflection padding) to make both dimensions multiples of 640 pixels; (2) the image is then tiled into a grid of non-overlapping 640 × 640 sub-images; (3) for each bounding box that overlaps a given sub-image, a clipped version is created that fits within the sub-image boundaries (coordinates are adjusted relative to the sub-image); (4) sub-images containing no remaining bounding boxes or where all clipped bounding boxes cover a negligible area (less than 1% of the sub-image) are automatically discarded. The resulting sub-images constituted the final dataset. When cutting images, overlapping bounding boxes were also split in the same manner. The algorithm automatically removed sub-images that did not contain any bounding boxes or had a negligible bounding box area. The final dataset consisted of 5025 images generated from an initial pool of 865 images.

The 640 × 640 resolution was selected because it is the default and recommended input size for most recent YOLO models in the Ultralytics implementation (including v5–v11 and RT-DETR), allowing direct training without additional resizing or sampling that could distort aspect ratios or introduce unnecessary padding artifacts. This size strikes an effective balance between preserving fine-grained details of distant/small fruits and maintaining computational efficiency during training and inference.

Path to CLIP-check solution. Once all images were correctly annotated, a comprehensive check was conducted on all resulting sub-images. One challenge was finding a way to detect sub-images with incorrect bounding boxes. Indeed, even after the automated check performed by CUT-IMG, some sub-images still contained bounding boxes that did not contain any oranges. This issue was mainly caused by the visual complexity of orchard scenes, where small fruits, occlusions, blur, and

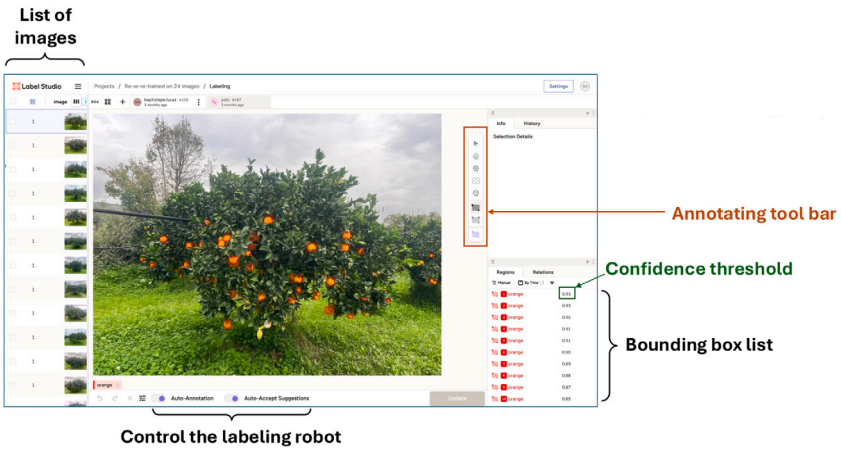

Fig. 4. Screenshot of the Label Studio framework used during the annotation process. Key interface components such as the image list, bounding box controls, confidence threshold slider, and robot labeling tools are highlighted.

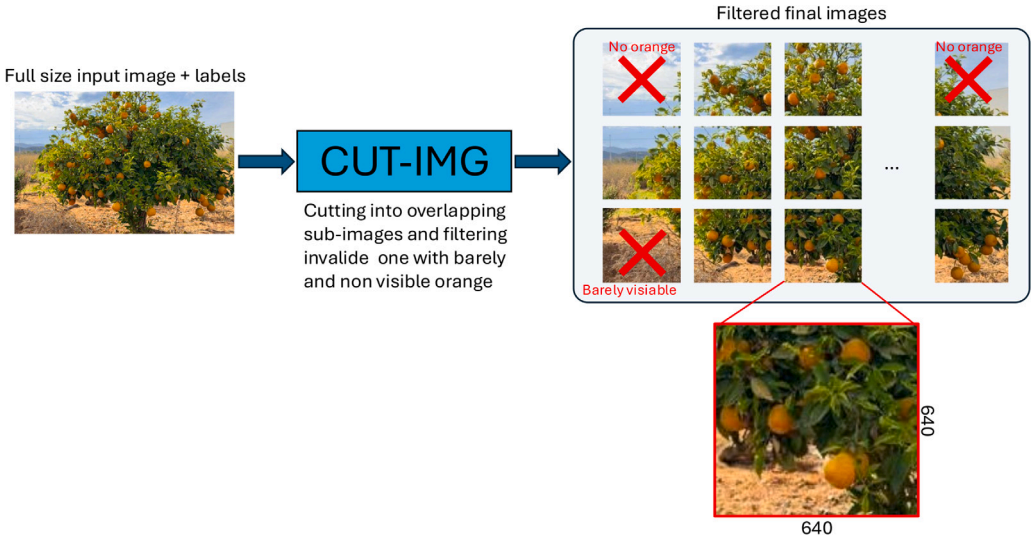

Fig. 5. Visualization of the CUT-IMG algorithm used for the computation of the final (640 × 640) images and filtering of invalid cases such as no orange in the image or barely visible ones.

texture similarity between oranges, leaves, branches, and background regions could still produce ambiguous or incorrect annotations after the geometric CUT-IMG step. Therefore, CUT-IMG ensured spatial consistency of the sub-images and clipped boxes, but it could not verify whether the visual content inside each remaining bounding box was semantically consistent with the target class.

One possible approach was to remove sub-images with bounding boxes whose proportional area (i.e. the bounding box area relative to the 640 × 640 total area) was too small. However, since many oranges occupied a very small proportional area, their bounding boxes would have been incorrectly removed. Another approach was to filter out bounding boxes with a high aspect ratio, but this too would have led to the removal of many oranges that were only partially visible in the images, resulting in tall-thin or short-wide bounding boxes.

An alternative solution was to create a 'background' dataset consisting of similar images without any oranges. A ViT model (Dosovitskiy et al., 2021) (google/vit-base-patch16-224-in21k) was then trained directly on the cropped bounding box images (padded to standardize the image format); however, the model still struggled to classify them as image distribution and resolution was varying greatly. Hence, a zero-shot classifier, named CLIP (Radford et al., 2021) was adopted to address this issue. The main challenge was to provide an exhaustive list of prompts covering all types of images. A total of 17 prompts were used, divided into two classes: the first six described images containing fruits, while the remaining prompts described images without fruits, considered as background (Table 3).

CLIP is an easy-to-use model based on textual representation of images combined with Vision Transformers (ViT). It allows the classification of any type of image since it has learned to recognize a wide variety of visual concepts and associate them with their names. It was trained by retrieving, for each image in the training dataset, the most probable textual prompt among a set of 32,768 randomly sampled text snippets. This was done using contrastive learning. These choices improved both performances and training efficiency: their best model was trained on 256 GPUs for two weeks. It can then be used in a zero-shot approach : providing a list of prompts sufficiently exhaustive to cover all types of images in the dataset, and CLIP will assign a probability to each prompt.

2.4. Object detection models and training protocol

The dataset was evaluated by fine-tuning several state-of-the-art object detection models from the You Only Look Once (YOLO) family, implemented using the Ultralytics library. Specifically, the models retained for benchmarking included YOLOv5mu, YOLOv5xu (Jocher, 2020), YOLOv8x (Jocher et al., 2023a), YOLOv10x (A. Wang et al.,

Table 2
Computational infrastructure used during the training and evaluation of object detection models. Details on GPUs, CPU, RAM, operating system, and key software frameworks such as PyTorch, Ultralytics, and CUDA versions are described.

Hardware	Software
GPUs: NVIDIA RTX A6000 48 GB × 3	Ubuntu 20.04.6 LTS
CPU: Intel(R) Core(TM) i9-10980XE @ 3.00 GHz	Python 3.10.15
Memory: 128 GB	PyTorch 1.13.1
	Ultralytics 8.3.9
	CUDA 12.4

2024), YOLO11n and YOLO11x (Jocher and Qiu, 2024). Concerning YOLOv8x and YOLO11x, both architectures were trained using a cosine learning rate scheduler, following a cosine curve over epochs. Performance results from two versions of the RT-DETR (Y. Zhao et al., 2024) (Real-Time Detection Transformer) were also included to enhance the benchmark: large (l) and extra-large (x).

These models were selected to cover a broad spectrum of the YOLO architecture evolution, from earlier iterations like YOLOv5, widely adopted in agricultural detection tasks (e.g., Song et al. (2025)), to more recent versions such as YOLOv10 and YOLOv11, which incorporate architectural improvements such as C2f modules and enhanced neck designs (Jegham et al., 2025). YOLOv11 in particular has been shown to maintain a favorable trade-off between speed and accuracy in high-density object scenarios (Jegham et al., 2025). To complement this spectrum, the standard versions of RT-DETR-l and RT-DETR-x were also evaluated in order to explore the behavior of transformer-based detectors in agricultural conditions, where occlusions, small objects, and complex textures are frequent. Recent work by Wu et al. (2025) proposed an enhanced RT-DETR variant for fruit ripening detection, achieving increased accuracy and reduced computational cost by integrating RepBlocks and multi-scale attention mechanisms. Although such modifications were not applied in this study, the inclusion of RT-DETR baselines enabled a meaningful comparison with convolutional YOLO models in terms of precision, recall, and inference performance.

As mentioned earlier, all models were downloaded from the Ultralytics library and pre-trained on the COCO (Lin et al., 2015) dataset.

All runs were conducted on the same data split: 70% for training, with the remaining 30% evenly split between the test and validation sets. In most cases, it was used a dropout rate of 0.2, which helped reduce training time while maintaining similar performance (around 3 h of training on 3 GPUs with a batch size of 12 and a maximum of 300 epochs). The patience parameter was set to 20 and the learning rate was varied from 0.01 to 0.0001.

2.5. Detection benchmark

Detection models were benchmarked after being trained on the 5025 sub-images. All training and evaluation procedures were carried out using the computational infrastructure detailed in Table 2.

This configuration ensured sufficient computational capacity for model fine-tuning and evaluation while maintaining reproducibility across experiments. The benchmark included YOLO (Redmon et al., 2016) models and one RT-DETR (Y. Zhao et al., 2024) model, all downloaded from the Ultralytics library (Jocher et al., 2023b). Their performance was compared using standard evaluation metrics described below, which follow standard definitions commonly adopted in object detection benchmarks (Tamilarasi et al., 2025).

Before evaluating precision and recall, it was necessary to define what constitutes a true positive (TP) in the context of object detection. In our case, the model predicted a bounding box, i.e., a rectangular region, intended to localize a fruit. A predicted box was considered a true positive only if it sufficiently overlapped with the corresponding ground truth annotation. This overlap was quantitatively assessed using the standard metric *Intersection over Union* (IoU), defined as:

$$\mathrm{IoU}(A_P, A_G) = \frac{|A_P \cap A_G|}{|A_P \cup A_G|} \tag{1}$$

where A_P and A_G denote the predicted and ground truth areas, respectively. The IoU quantifies the degree of spatial agreement between the predicted and true bounding boxes. A prediction was classified as a true positive if IoU > 0.5, following common practice in object detection benchmarks. Predictions failing to meet this threshold were considered either false positives (FP) if no corresponding object was present, or false negatives (FN) if the object was not detected.

Based on this, conventional evaluation metrics were derived. Precision corresponds to the proportion of correct detections over all predictions:

$$\mathrm{Precision} = \frac{TP}{TP + FP} \tag{2}$$

Recall quantifies the ability of the model to retrieve all relevant objects:

$$\mathrm{Recall} = \frac{TP}{TP + FN} \tag{3}$$

Since precision and recall often exhibit a trade-off, the F1 score — defined as the harmonic mean of the two — offers a balanced measure of overall detection performance:

$$\mathrm{F1} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{4}$$

True negatives (TNs) are typically excluded in object detection tasks, as the vast majority of image regions represent background and do not correspond to any annotated object.

2.6. Data records

The final data was divided into nine subfolders, corresponding to nine weather conditions (WC). Each WC is made of two letters. The first one describes the time of the day : morning (M), afternoon (A), evening (E). The second one describes the weather : sunny (S), cloudy (C), rainy (R). Due to the absence of rainy conditions in the morning and evening, the dataset includes night (NI) and indoor (IN) categories instead of defining specific labels for morning rainy (MR) and evening rainy (ER) scenarios.

Each sub-image follows the naming convention in Fig. 6a. To change the data structure, it is necessary to ensure that the image is moved along with its associated label .txt file. In addition, the corresponding part in the annotations.json also needs to be moved (the image and label id from the .json file remain the same regardless the folder in which they are stored).

The orange dataset is available at Mendeley Data and contains a total of 5025 images for orange trees. Each tree has at least one orange, ranging in color from green to orange, and the dataset includes diverse weather conditions.

Once the dataset is downloaded and unpacked, it is organized into a hierarchical folder structure. At the top level, there is one folder for each of the nine WCs, named according to the two-letters WC convention described above. Within each WC subfolder, users will find the actual dataset files. This is the second and final level of the file structure, where three types of data are available:

- **Images** Each image can be identified by the *.jpg* extension. All images follow the naming convention described in Fig. 6.
- **Labels** Text files can be identified by the *.txt* extension. Each image has a corresponding text file with the same name, where only the extension is switched from *.jpg* to *.txt*. These files store the bounding boxes, with each line containing the coordinates of one bounding box, corresponding to a single orange in the associated image. The format of each bounding box follows the YOLO format, with each bounding box being represented by five values

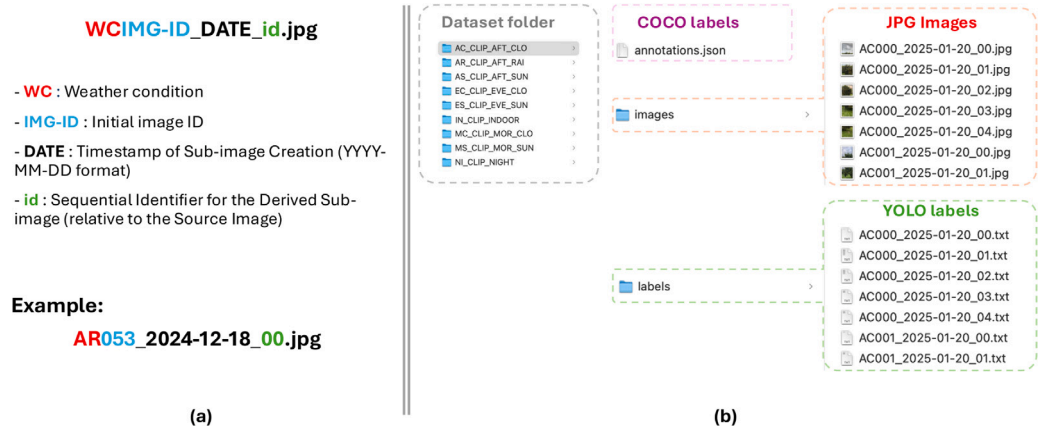

Fig. 6. Summary of data folder structuring and file conventions. Panel (a) illustrates the naming convention for images, which includes encoded information about the weather condition, original image ID, date, and sub-image index. Panel (b) shows the hierarchical folder structure used to organize the dataset, where each folder corresponds to a specific weather-time combination.

$[c_{ID}, x_{center}, y_{center}, width, height]$ where c_{ID} is the class label ID, x_{center} and y_{center} are the normalized coordinates of the center of the bounding box, and $width$ and $height$ are the normalized dimensions of the box.

- **annotations.json** This file allows to work with detection models following the COCO (Lin et al., 2015) format. Within the .json file, the image_id is generated by taking the number associated with each WC letter and removing the underscores and hyphens. For example, AR053_2024-12-18_00.jpg becomes 11805320241218 as A (resp. R) gives 1 (resp. 18).

This structured organization, along with consistent naming conventions, ensures that the data is easily accessible and ready for use in a variety of computer vision tasks (Fig. 6b).

2.7. Technical validation

A technical validation process was conducted to ensure the accuracy and reliability of the orange fruit dataset. This process consisted of three main experiments designed to evaluate the quality of the data and its applicability to further analysis and modeling:

(1) **CLIP check for orange content on sub-images**. CLIP (Radford et al., 2021) (Contrastive Language-Image Pre-Training) model was used to analyze sub-images resulting from the CUT-IMG algorithm to ensure dataset quality. CLIP has demonstrated strong performance in connecting visual content with textual descriptions. By querying it with prompts related to oranges and, more broadly, fruits, the objective was to determine whether each sub-image contained at least one orange or consisted of empty bounding boxes only.

(2) **Benchmark of detection models**. Performance evaluation was conducted on several object detection models fine-tuned using the orange fruit dataset. This involved training and testing the models on a subset of the data to measure their ability to accurately identify and localize oranges within images. The goal of this experiment was to identify the most effective detection model for the orange fruit dataset. This would enable us to subsequently use the best-performing model for downstream tasks such as counting oranges in images or assessing fruit quality.

(3) **Ablation study on dataset curation pipeline**. An ablation study was conducted to quantify the contribution of the proposed preprocessing steps, namely CUT-IMG and CLIP-based filtering. The analysis was performed using the best-performing model identified in the detection benchmark (YOLO10x), in order to isolate the effect of the dataset curation pipeline. The detector was trained under two controlled conditions: (i) using the original dataset of images and (ii) using the post-processed dataset obtained after applying CUT-IMG and CLIP filtering. Both models were evaluated on the same held-out subset sampled from the final dataset, ensuring consistency in acquisition conditions. Performance was assessed using standard object detection metrics, including precision, recall, mAP@0.5, and mAP@0.5:0.95, along with qualitative inspection of prediction outputs. This experimental design allows isolating the impact of the preprocessing pipeline on model performance while keeping all other factors constant.

2.8. CLIP classification

A list of 17 prompts was provided (Table 3), and fed them into CLIP. The first six prompts corresponded to the fruit category, without distinguishing between an orange or a lemon. The key point was that each image should be classified either as containing a fruit or as showing only the background, which is represented by the eleven remaining prompts.

This classification was conducted on all images based on two approaches:

Mean approach. This method computed the average probability among the first six prompts (corresponding to fruit-related prompts) and the remaining eleven prompts (corresponding to background-related prompts). The image was classified as "Fruit" if the fruit-related average probability was higher than the background-related one. These averages are denoted as *Fruit Mean* and *Backg Mean*.

Argmax approach. This method classified an image based on the prompt with the highest probability. If the highest probability (argmax) corresponded to one of the first six prompts, the image was classified as containing a fruit; otherwise, it was classified as background.

3. Results and discussion

The difference between the first and second highest probabilities was easily noticeable (Fig. 7(a)), ensuring the relevance of this approach, i.e. CLIP did not assign similar probabilities to all prompts, which can also be observed in Fig. 7(c). This confirmed the strong semantic discrimination capabilities of CLIP when applied to agricultural images, as also highlighted by Zhang et al. (2025), which emphasized the value of zero-shot classifiers in reducing manual annotation burden in complex datasets.

Finally, within each WC, the majority of images were classified as containing a fruit (Fig. 7(b)). This outcome resulted from the process described below. Initially, there were still more fruit images, but

Table 3
Full list of 17 text prompts employed in the zero-shot classification process using CLIP. The prompts are divided into two groups: six describing images that contain fruit (specifically oranges or lemons, ripe and unripe) and eleven that represent background or non-fruit content.

Label	Description
p0	A picture of an orange in a tree
p1	A picture of an unripe orange in a tree
p2	A picture of a part of an orange in a tree
p3	A picture of a part of an unripe orange in a tree
p4	A picture of a lemon in a tree
p5	A picture of an unripe lemon in a tree
p6	A picture of leaves and tree branches without any ripe or unripe fruit
p7	A picture of leaves and tree branches
p8	A picture of parts of leaves and tree branches
p9	A picture of parts of leaves and tree branches without any ripe or unripe fruit
p10	A picture of a building
p11	A picture of a part of a building
p12	A picture of a part of the sky
p13	A picture of dead leaves on the ground without any ripe or unripe fruit
p14	A picture of dead leaves on the ground
p15	A picture of the roots of a tree
p16	A picture of the roots of a tree without any ripe or unripe fruit

the number of background images was also higher, leading to larger orange-blue and green-blue gaps between bars. Images within these gaps were then reviewed (mostly duplicates) and those that, indeed, did not contain a fruit — or where the fruit was too small to be relevant — were removed from the dataset. Additionally, some images displayed fruits clearly, but there were also parts of larger bounding boxes that were not of interest. In such cases, the dataset was refined by removing those instances and retaining only the well-defined bounding boxes. All figures presented here were generated after reviewing and filtering out false-positive images. Therefore, while a small fraction of background images remains misclassified by CLIP (as the image content itself was unchanged), incorrect bounding boxes and irrelevant images were removed during the refinement process. Ultimately, the low average probability for each background prompt confirms that the dataset is of high quality (Fig. 7(c)). The combination of CLIP verification and manual filtering represents a novel, semi-automated approach to improving dataset consistency. This strategy is in line with recent trends in visual-language integration for classification and detection tasks, such as E-CLIP (Zhang et al., 2025) for fruit identification and YOLO-CLIP (An et al., 2025) models for multimodal recognition in unstructured environments.

From an initial dataset of 5742 images, a CLIP-based filtering process was implemented to eliminate false positives, resulting in a refined set of 5025 images. While CLIP has so far been primarily integrated into detection architectures to enable open-vocabulary prediction or pseudo-labeling workflows (X. Wang et al., 2024; Li et al., 2025), in this study it was used to externally as a zero-shot filter for dataset cleaning, an approach that preserves detection model simplicity while achieving semantic consistency. This refinement step aimed to enhance the accuracy and relevance of the image data for subsequent analysis. Subsequent analysis of the refined image set involved summarizing the properties of the bounding boxes used to locate the oranges in the field. In this context, the analysis focused on characterizing the distribution and properties of these bounding boxes across the entire image set. Overall, A total of 43,038 bounding boxes were identified across all images. This substantial number of bounding boxes suggests a rich set of objects or regions of interest within the images. One main property of these bounding boxes is that each one has an even-numbered width and height dimension (Fig. 7(d)). Specifically, the heatmap reveals distinct vertical and horizontal bands that correspond to bounding boxes with even-numbered width and height values, an expected outcome of the CUT-IMG tiling process. This regularity confirms the systematic structure of the dataset and may facilitate more stable convergence during model training by minimizing shape irregularities. The geometric consistency observed here aligns with recent insights from bounding box regression literature. Zhang and Zhang (2024) emphasize how the distribution of bounding box shapes and scales influences localization performance, suggesting that uniformity in size and aspect ratio can simplify the regression task and reduce variance across training samples. Likewise, Ravi et al. (2022) demonstrate that their BIoU loss performs better in the presence of regular and symmetric bounding boxes, conditions that naturally emerge in our dataset due to its preprocessing strategy.

Object detection

Details on the object detection models and training protocol are provided in Section 2.6. A benchmark comparing the performance of these models using standard detection metrics, described in Fig. 4, was provided below. A clear difference emerged between RT-DETR and YOLO-based models: the former tended to predict significantly more oranges (i.e. a higher TP count), but this came at the cost of increased false positives, as it often misclassified background as oranges. As a result, RT-DETR achieved a higher recall but lower precision (Fig. 8(a)).

Besides performing worse than YOLO models, RT-DETR models were also significantly slower during inference, even though they are classified as real-time detectors (Fig. 8(b)). The frames per second (FPS) benchmark was conducted using a single GPU, with a batch size of 12 images, on an 11-second test video. An additional observation was made at the WC level (Fig. 8(c)): by structuring the data based on weather conditions, a benchmark of detection metrics was established during the validation stage of the best fine-tuned model, namely YOLOv10x. The results for the IN category were worse since it contains only 18 images, which depict open oranges, whereas all other images show fruits on trees. This was done by using a batch size of 12 images with a confidence threshold set to 0.5, inferring on each of the three GPUs. These results highlight how transformer-based detectors like RT-DETR, while theoretically suited for open-world and long-range dependencies, may be less effective than convolutional models in domain-specific agricultural tasks, particularly when constrained by limited hardware or real-time requirements. This outcome aligns with recent findings by Wu et al. (2025), where RT-DETR architectures required custom optimization (e.g., RepBlocks) to compete with YOLO variants in fruit ripening evaluation tasks. In our case, the superior trade-off between speed and accuracy achieved by YOLOv10x reinforces the evidence that recent YOLO models, especially those adopting architectural advances like C2f modules and dynamic heads, are better suited for detection in complex field conditions (Jegham et al., 2025; Alif and Hussain, 2024). The weather-specific evaluation further confirms the robustness of YOLOv10x, which maintained high F1-scores across all illumination and background contexts, validating the generalization capabilities of both the model and the dataset itself.

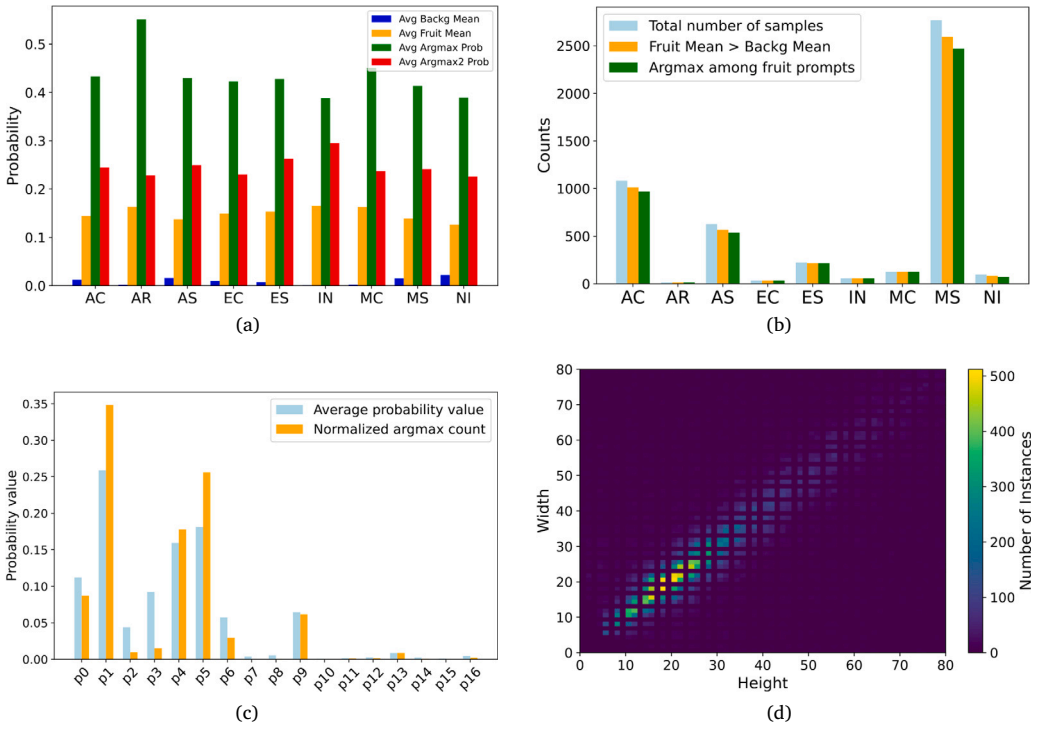

Fig. 7. Key validation metrics from the CLIP-based image filtering process. Panel (a) shows average CLIP classification probabilities across different weather conditions. Panel (b) reports the number of sub-images classified as containing fruit using two different classification strategies. Panel (c) summarizes average classification probabilities for each prompt used with CLIP, indicating the model's ability to distinguish between fruit and background. Panel (d) provides a 2D histogram of the bounding box dimensions (width and height).

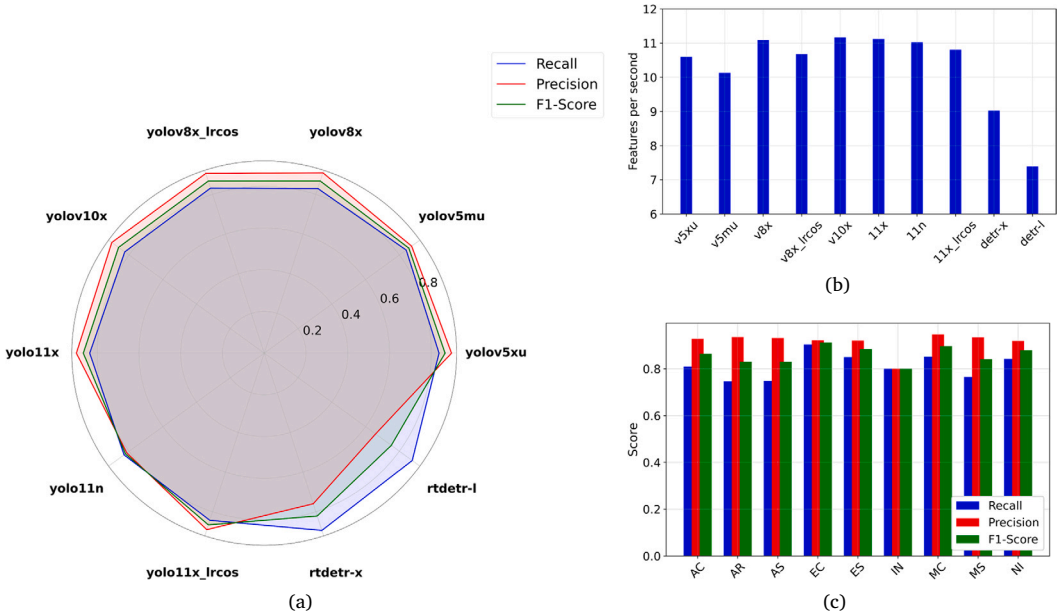

Fig. 8. Benchmark results of multiple object detection models trained on the dataset. Panel (a) compares model performance using standard metrics: precision, recall, and F1-score. Panel (b) illustrates inference speed in FPS for each model. Panel (c) provides a breakdown of detection performance across different weather conditions using the best-performing model (YOLOv10x). These results confirm the effectiveness of the dataset in supporting accurate and efficient object detection in variable field conditions.

The superiority of recent YOLO models (particularly YOLOv10 and YOLOv11) over RT-DETR on this dataset can be attributed to several factors rooted in architectural differences, the specific challenges of the dataset, and optimization considerations. YOLO models employ a single-stage, CNN-based architecture with efficient backbones and, in later versions, Efficient Layer Aggregation Networks, which excel at hierarchical feature extraction from local patterns. In our dataset, oranges frequently appear in dense clusters with heavy occlusions from leaves,

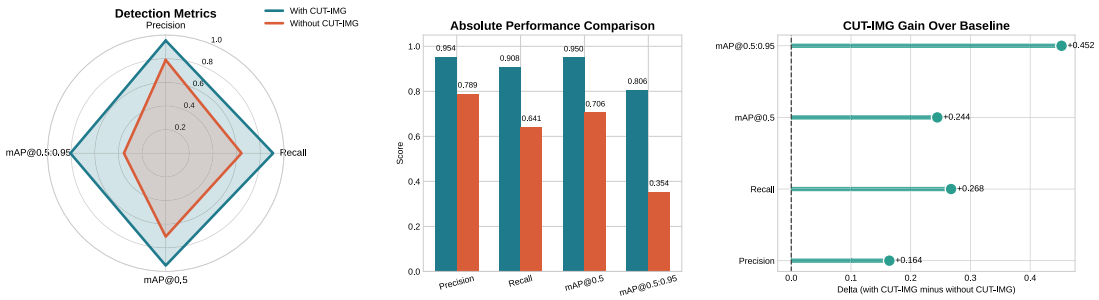

Fig. 9. Ablation study comparing the detector trained with CUT-IMG and the baseline trained without CUT-IMG on the same held-out evaluation subset. The radar plot, grouped bar chart, and delta plot consistently show better precision, recall, mAP@0.5, and mAP@0.5:0.95 when CUT-IMG is included, confirming its positive contribution to overall detection performance.

branches, and netting under highly variable lighting and weather conditions. YOLO's path aggregation mechanisms and multi-scale feature fusion are particularly effective for detecting small and partially visible fruits, resulting in higher overall accuracy (YOLOv11m achieved mAP@0.5 of 0.892 vs. RT-DETR's 0.851) and better recall in occluded scenarios. In contrast, RT-DETR, as a transformer-based detector, relies heavily on attention mechanisms to capture global context, which can be advantageous in less cluttered scenes but introduces computational overhead and sensitivity to background noise in dense agricultural environments. This often leads to over-attention to irrelevant features such as textured leaves or bark, leading to an increase in false positives and reducing precision. Regarding inference speed, YOLO architectures are designed from the ground up for real-time performance with lightweight convolutional operations, whereas transformers involve significantly more parameters and quadratic complexity in attention layers, resulting in slower inference.

Ablation study

To quantify the contribution of CUT-IMG combined to CLIP filtering, an ablation study was performed in which the detector YOLO10x was trained with post-processed images using the same hyperparameters (100 epochs, Learning rate 0.01 and image size of 640 × 640) while seeding to 42 for reproducibility and then was compared against a baseline trained on the original dataset of images. Both models were tested on the same subset sampled from the final dataset across all acquisition conditions. As shown in Fig. 9, the model trained with the post-processing steps consistently outperformed the baseline in all main detection metrics, including precision, recall, mAP@0.5, and mAP@0.5:0.95. The gains indicate that these additional processing steps improved both detection reliability and localization quality.

The qualitative analysis shows where the CUT-IMG + CLIP filtering model outperforms the original images (examples of the predictions in Fig. 10). In the several examples, the version trained with CUT-IMG and CLIP filtering produces fewer false negatives and false positives, while also achieving better bounding box approximation. In other extreme cases, the mean IoU increases from 0.706 on originals to 0.95, which correspond to an absolute gain of 0.244. Over the same samples, the average number of false negatives decreases from 16.9 to 3.9 per image, and the average number of false positives decreases from 21.5 to 1.9 per image.

The observed improvements can be attributed to the complementary roles of the two processing steps. CUT-IMG reduces scene complexity by focusing the model on localized regions, effectively increasing the relative size of fruit instances and reducing background noise. This facilitates feature learning and improves both detection sensitivity and localization accuracy. In parallel, CLIP-based filtering enhances dataset quality by removing ambiguous or low-informative samples, thus reducing label noise and improving the consistency of the training data. The combined effect of these steps leads to a dataset that is both more informative and less noisy, which directly translates into improved model generalization. These findings are consistent with previous studies showing that performance improvements in fruit detection models are often achieved through the combined contribution of multiple components rather than a single modification. For instance, Lawal et al. (2021) demonstrated through ablation experiments that incremental changes in backbone design, activation functions, and feature aggregation strategies can lead to measurable improvements in detection accuracy and robustness under real-world conditions. Similarly, recent works have highlighted the importance of data quality and feature selection in improving model performance. In vision-language models, selective filtering and combination of local and global features have been shown to enhance classification accuracy and generalization, particularly under heterogeneous conditions (Cao et al., 2025). These findings support the idea that removing low-informative or misleading visual information can significantly improve downstream tasks. In this context, the approach adopted in this study emphasizes simplicity, scalability, and applicability under real-world field conditions. Rather than relying on complex data curation strategies, the proposed pipeline leverages straightforward yet effective preprocessing steps that can be easily integrated into practical workflows. Overall, the results suggest that, beyond methodological sophistication, the effectiveness of dataset curation pipelines strongly depends on their ability to improve data quality and reduce noise in realistic acquisition scenarios. Recent work has proposed more advanced and principled data curation strategies, such as clustering-based approaches for self-supervised filtering (Vo et al., 2024). While these methods provide a more formalized framework for dataset refinement, they typically involve higher computational complexity and are not specifically tailored to real-field agricultural scenarios. In contrast, the approach adopted in this study prioritizes simplicity, scalability, and ease of integration into practical workflows, which are key requirements for operational deployment in agricultural environments. This is particularly relevant in agricultural environments, where variability in lighting, occlusion, and background complexity represents a major challenge for robust fruit detection.

Limitations

While the proposed dataset was designed to maximize environmental and acquisition variability, some limitations should be acknowledged. First, although images were collected under diverse lighting conditions, orchard structures, and fruit ripening stages, data acquisition was geographically restricted to southern Italy. This intra-regional variability mitigates, but does not fully eliminate, potential regional bias when generalizing to citrus-growing systems with substantially different pedoclimatic, agronomic and landscape conditions (e.g., canopy architecture, background composition, and orchard layout). Second, all images were acquired using consumer-grade smartphone cameras. This choice reflects realistic deployment scenarios for fruit detection applications and does not limit detection performance. However, the

Fig. 10. Examples from the ablation analysis illustrating that the detector trained with CUT-IMG + CLIP filtering yields better alignment with fruit instances, fewer missed detections, and fewer false positive predictions than the detector trained on original dataset. The green bounding boxes refer to the ground truth and red are the predictions.

dataset was designed for RGB-based visual analysis and is therefore not intended for applications requiring calibrated color or geometric measurements, which would require additional sensing modalities or calibration procedures beyond the scope of the present work. Third, the dataset adopts a single-class annotation strategy focused exclusively on orange fruits. This represents a deliberate design choice aligned with the intended detection and ripening assessment tasks, but it does not support multi-species or multi-task learning without further annotation. Finally, while the CUT-IMG segmentation procedure is effective in standardizing input resolution and improving annotation efficiency, it may partially reduce spatial context in a limited number of edge cases, such as objects located near segment boundaries. These effects have been mitigated through manual review and CLIP-based filtering, but remain an inherent compromise of tile-based preprocessing strategies.

4. Conclusion

In this work, a large-scale, multi-condition dataset for orange detection and ripening assessment in citrus orchards was introduced. The dataset was designed to reflect high variability in environmental conditions and device characteristics. A CLIP-based filtering pipeline ensured label quality and relevance, while benchmark experiments confirmed that YOLOv10x performs best in terms of accuracy and speed. This dataset holds promise for future applications in fruit ripening estimation, yield forecasting, and real-time orchard monitoring. Future work will explore expanding the dataset to other fruit species and integrating 3D sensing for size estimation.

CRediT authorship contribution statement

Alessandro Carella: Writing – review & editing, Writing – original draft, Visualization, Investigation, Formal analysis, Data curation. **Baptiste Paul Ernest Lucas:** Writing – review & editing, Writing – original draft, Software, Investigation, Formal analysis, Data curation. **Safouane El Ghazouali:** Writing – review & editing, Validation, Investigation, Formal analysis, Data curation. **Pedro Tomas Bulacio Fischer:** Writing – review & editing, Data curation. **Roberto Massenti:** Methodology, Investigation, Formal analysis, Data curation. **Francesca Venturini:** Methodology, Investigation, Formal analysis, Conceptualization. **Umberto Michelucci:** Writing – review & editing, Supervision, Resources, Methodology, Investigation, Formal analysis, Conceptualization. **Riccardo Lo Bianco:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization.

Code availability

The dataset described in this study is available via Mendeley Data. The implementation of the experiments, along with the corresponding results, can be accessed in the public GitHub repository: https://github.com/toelt-llc/UNIPA-oranges. All software dependencies and version specifications are documented in the repository's **README.md** and **environment.yml** files. Additionally, detailed instructions for the annotation workflow using Label Studio are provided in the file **label-studio.md** within the same repository.

Funding

The present study was funded by the European Union's Horizon 2020 research and innovation programme, under the Marie Skłodowska-Curie RISE action, project "SUSTAINABLE", grant agreement No. 101007702.

Declaration of competing interest

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

Acknowledgments

The authors would like to thank Salvatore Bacino, Alessandro Cavarretta, Nicola Pizzolato, and Michele Ferrantelli for their contribution in providing a significant portion of the field images used in this dataset.

Data availability

I have shared the link to my data/code at the attach file step

Oranges in the field (Original data) (Mendeley Data)

References

Alif, M.A.R., Hussain, M., 2024. YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain. arXiv preprint. https://doi.org/10.48550/arXiv.2406.10139.

An, J., Su, P., Liu, J., Li, G., 2025. Implementation of YOLO-CLIP fusion algorithm for fall detection. Signal Image Video Process. 19 (10), 837, https://doi.org/10.1007/s11760-025-04397-w.

Arakawa, T., Tanaka, T.S.T., Kamio, S., 2024. Detection of on-tree chestnut fruits using deep learning and RGB unmanned aerial vehicle imagery for estimation of yield and fruit load. Agron. J. 116 (3), 973–981, https://doi.org/10.1002/agj2.21330.

Bonora, A., Bortolotti, G., Bresilla, K., Grappadelli, L.C., Manfrini, L., 2021. A convolutional neural network approach to detecting fruit physiological disorders and maturity in 'Abbé Fétel' pears. Biosyst. Eng. 212, 264–272, https://doi.org/10.1016/j.biosystemseng.2021.10.009.

Bortolotti, G., Piani, M., Gullino, M., Mengoli, D., Franceschini, C., Grappadelli, L.C., Manfrini, L., 2024. A computer vision system for apple fruit sizing by means of low-cost depth camera and neural network application. Precis. Agric. 25 (6), 2740–2757, https://doi.org/10.1007/s11119-024-10139-8.

Cao, Y., Xing, S., Yu, Z., Wu, C., Weng, Z., Du, J., 2025. An optimal feature selection fusion method of visual models for CLIP. In: 2025 44th Chinese Control Conference. CCC, pp. 8833–8838, https://doi.org/10.23919/CCC64809.2025.11179370.

Carella, A., Massenti, R., Lo Bianco, R., 2023. Testing effects of vapor pressure deficit on fruit growth: A comparative approach using peach, mango, olive, orange, and loquat. Front. Plant. Sci. 14, 1294195, https://doi.org/10.3389/fpls.2023.1294195.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint. https://doi.org/10.48550/arXiv.2010.11929.

Dwyer, B., Nelson, J., Hansen, T., et al., 2024. Roboflow. Computer vision software platform, version 1.0. https://roboflow.com.

James, J.A., Manching, H.K., Mattia, M.R., Bowman, K.D., Hulse-Kemp, A.M., Beksi, W.J., 2024. CitDet: A benchmark dataset for citrus fruit detection. IEEE Robot. Autom. Lett. 9 (12), 10788–10795, https://doi.org/10.1109/LRA.2024.3474473.

Jegham, N., Koh, C.Y., Abdelatti, M., Hendawi, A., 2025. YOLO evolution: A comprehensive benchmark and architectural review of YOLOv12, YOLO11, and their previous versions. ArXiv preprint. https://doi.org/10.48550/arXiv.2411.00201.

Jocher, G., 2020. Ultralytics YOLOv5. http://dx.doi.org/10.5281/zenodo.3908559, Version 7.0, AGPL-3.0 license. https://github.com/ultralytics/yolov5.

Jocher, G., Chaurasia, A., Qiu, J., 2023a. Ultralytics YOLOv8. Version 8.0.0, AGPL-3.0 license. https://github.com/ultralytics/ultralytics.

Jocher, G., Chaurasia, A., Qiu, J., Autin, A., Nelson, J., et al., 2023b. Ultralytics YOLOv5, YOLOv8 and vision AI models. https://github.com/ultralytics/ultralytics. (Accessed 28 May 2025).

Jocher, G., Qiu, J., 2024. Ultralytics YOLO11. Version 11.0.0, AGPL-3.0 license. https://github.com/ultralytics/ultralytics.

Kamilaris, A., Prenafeta-Boldú, F.X., 2018. Deep learning in agriculture: A survey. Comput. Electron. Agric. 147, 70–90, https://doi.org/10.1016/j.compag.2018.02.016.

Lawal, O.M., Huamin, Z., Fan, Z., 2021. Ablation studies on yolofruit detection algorithm for fruit harvesting robot using deep learning. IOP Conf. Ser Earth Env. Sci. 922 (1), 012001. http://dx.doi.org/10.1088/1755-1315/922/1/012001, https://doi.org/10.1088/1755-1315/922/1/012001.

Li, J., Sun, S., Zhang, K., Zhang, J., Zhuo, L., 2025. Single-stage zero-shot object detection network based on CLIP and pseudo-labeling. Int. J. Mach. Learn. Cybern. 16 (2), 1055–1070, https://doi.org/10.1007/s13042-024-02321-1.

Lin, Y., Huang, Z., Liang, Y., Liu, Y., Jiang, W., 2024. AG-YOLO: A rapid citrus fruit detection algorithm with global context fusion. Agriculture 14 (1), https://doi.org/10.3390/agriculture14010114.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P., 2015. Microsoft COCO: Common objects in context. arXiv preprint. https://doi.org/10.48550/arXiv.1405.0312.

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.

Massenti, R., Lo Bianco, R., Sandhu, A.K., Gu, L., Sims, C., 2016. Huanglongbing modifies quality components and flavonoid content of 'Valencia' oranges. J. Sci. Food Agric. 96 (1), 73–78, https://doi.org/10.1002/jsfa.7061.

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

Mossad, A., Farina, V., Lo Bianco, R., 2020. Fruit yield and quality of 'Valencia' orange trees under long-term partial rootzone drying. Agronomy 10 (2), https://doi.org/10.3390/agronomy10020164.

Nawaz, U., Awais, M., Gani, H., Naseer, M., Khan, F., Khan, S., Anwer, R.M., 2024. AgriCLIP: Adapting CLIP for agriculture and livestock via domain-specialized cross-model alignment. arXiv preprint. https://doi.org/10.48550/arXiv.2410.01407.

Oviedo Espinosa, M.R., Porto, L.R., Orlando, V.S.W., Tommaselli, A.M.G., Dal Poz, A.P., Imai, N.N., 2024. Evaluation of YOLO efficiency in automatic orange detection in multi-exposure images. ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci. X-3-2024, 303–308, https://doi.org/10.5194/isprs-annals-X-3-2024-303-2024.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I., 2021. Learning transferable visual models from natural language supervision. ArXiv preprint. https://doi.org/10.48550/arXiv.2103.00020.

Ravi, N., Naqvi, S., El-Sharkawy, M., 2022. BIoU: An improved bounding box regression for object detection. J. Low Power Electron. Appl. 12 (4), 51, https://doi.org/10.3390/jlpea12040051.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. arXiv preprint. https://doi.org/10.48550/arXiv.1506.02640.

Shi, R., Li, T., Yamaguchi, Y., 2020. An attribution-based pruning method for real-time mango detection with YOLO network. Comput. Electron. Agric. 169, 105214, https://doi.org/10.1016/j.compag.2020.105214.

Song, J., Kim, D., Jeong, E., Park, J., 2025. Determination of optimal dataset characteristics for improving YOLO performance in agricultural object detection. Agriculture 15 (7), 731, https://doi.org/10.3390/agriculture15070731.

Tamilarasi, T., Muthulakshmi, P., Miraei Ashtiani, S.-H., 2025. Improved YOLO-based real-time brinjal detection algorithm for vision modules in harvesting robots. Eng. Res. Express 7 (3), 035234, https://doi.org/10.1088/2631-8695/adef00.

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

Vo, H.V., Khalidov, V., Darcet, T., Moutakanni, T., Smetanin, N., Szafraniec, M., Touvron, H., Couprie, C., Oquab, M., Joulin, A., Jégou, H., Labatut, P., Bojanowski, P., 2024. Automatic data curation for self-supervised learning: A clustering-based approach. arXiv:2405.15613. https://doi.org/10.48550/arXiv.2405.15613.

Wang, D., Cao, W., Zhang, F., Li, Z., Xu, S., Wu, X., 2022. A review of deep learning in multiscale agricultural sensing. Remote. Sens. 14 (3), 559, https://doi.org/10.3390/rs14030559.

Wang, A., Chen, H., Liu, L., et al., 2024. YOLOv10: Real-time end-to-end object detection. arXiv preprint. https://doi.org/10.48550/arXiv.2405.14458.

Wang, X., Ren, W., Chen, X., Fan, H., Tang, Y., Han, Z., 2024. Uni-YOLO: Vision-language model-guided YOLO for robust and fast universal detection in the open world. In: Proceedings of the 32nd ACM International Conference on Multimedia. MM'24, Association for Computing Machinery, New York, NY, USA, pp. 1991–2000, https://doi.org/10.1145/3664647.3681212.

Wu, H., Mo, X., Wen, S., Wu, K., Ye, Y., Wang, Y., Zhang, Y., 2024. DNE-YOLO: A method for apple fruit detection in diverse natural environments. J. King Saud Univ. Comput. Inf. Sci. 36 (9), 102220, https://doi.org/10.1016/j.jksuci.2024.102220.

Wu, M., Qiu, Y., Wang, W., Su, X., Cao, Y., Bai, Y., 2025. Improved RT-DETR and its application to fruit ripeness detection. Front. Plant Sci. 16, 1423682, https://doi.org/10.3389/fpls.2025.1423682.

Xiao, X., Wang, Y., Jiang, Y., Wu, H., Zhang, Z., Wang, R., 2024. AC-YOLO: Citrus detection in the natural environment of orchards. J. Agric. Eng. 55 (4), https://doi.org/10.4081/jae.2024.1654.

Zhang, Y., Shao, Y., Tang, C., Liu, Z., Li, Z., Zhai, R., Peng, H., Song, P., 2025. E-CLIP: An enhanced CLIP-based visual language model for fruit detection and recognition. Agriculture 15 (11), 1173, https://doi.org/10.3390/agriculture15111173.

Zhang, H., Zhang, S., 2024. Shape-IoU: More accurate metric considering bounding box shape and scale. arXiv preprint. https://arxiv.org/abs/2312.17663.

Zhang, L., Zhou, G., Chen, A., Yu, W., Peng, N., Chen, X., 2023. Rapid computer vision detection of apple diseases based on AMCFNet. Multimedia Tools Appl. 82 (29), 44697–44717, https://doi.org/10.1007/s11042-023-15548-x.

Zhao, J., Du, C., Li, Y., Mudhsh, M., Guo, D., Fan, Y., Wu, X., Wang, X., Almodfer, R., 2024. YOLO-Granada: A lightweight attentioned yolo for pomegranates fruit detection. Sci Rep 14 (1), 16848, https://doi.org/10.1038/s41598-024-67526-4.

Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., Chen, J., 2024. DETRs beat YOLOs on real-time object detection. In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 16965–16974, https://doi.org/10.1109/CVPR52733.2024.01605.