Computers and Electronics in Agriculture 218 (2024) 108728

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Original papers

Optimizing tomato plant phenotyping detection: Boosting YOLOv8 architecture to tackle data complexity

Firozeh Solimani [a], Angelo Cardellicchio [a,*], Giovanni Dimauro [b], Angelo Petrozza [c], Stephan Summerer [c], Francesco Cellini [c], Vito Renò [a]

[a] Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, National Research Council of Italy, Via Amendola 122, D/O, Bari, 70126, Italy
[b] University of Bari, Department of Computer Science, Via E. Orabona, 4, Bari, 70125, Italy
[c] ALSIA Centro Ricerche Metapontum Agrobios, s.s. Jonica 106, km 448.2, Metaponto, 75010, Italy

ARTICLE INFO

Keywords:
Tomato plant phenotyping
YOLOv8
Nodes
Fruit
Flower identification

ABSTRACT

Effective identification of tomato plant traits is crucial for timely monitoring and evaluating their growth and harvest. However, conducting stress experiments on multiple tomato genotypes introduces challenges due to the nature of the data. One of these challenges arises from an imbalanced sample distribution, potentially leading to misclassification between classes and disruptions in model recognition. This paper addresses the effect of these challenges by considering the imbalanced classes of flowers, fruits, and nodes and proposing an improved detection approach through data balancing. A novel data-balancing approach is introduced in this study to overcome the issue of imbalanced data. The proposed solution involves the implementation of a YOLOv8 deep learning model, which effectively detects flowers, fruits, and nodes in tomato plants. This model significantly enhances the ability of the algorithm to detect objects of varying sizes within complex environments. To further bolster the recognition capability of the targeted classes, the proposed model integrates a Squeeze-and-Excitation (SE) block attention module into its head architecture. This module strengthens the model recognition ability by giving increased attention to the studied classes, thereby enhancing overall detection performance. The results demonstrate that the data balancing approach successfully improves the model performance in response to the data challenges. When applying the technique of pre-training the optimal weights obtained from balanced data on imbalanced data, the SE-block module showed significant improvements in outcomes.

1. Introduction

Recent catastrophic events caused by climate change and the constantly growing population led to the need to achieve the optimal trade-off between crop yields and environmental sustainability. As such, the detailed analysis of plant morphological traits (e.g., nodes, fruit, or flowers) plays a central role in providing critical information on aspects such as the span of the growth period and the optimal time for flowering and fruit harvesting (Maji et al., 2022). This aspect was further stressed by the developments achieved in *high-throughput phenotyping* (HTP), an entirely new field related to the provisioning of high quantities of data able to characterize the phenotyping traits of plants. The amount of data available to the researchers allowed them to address a whole set of new applications, mainly related to the real-time identification of phenotypical traits under challenging conditions (Bac et al., 2017; Afonso et al., 2020; Cardellicchio et al., 2023).

One of the most studied crops is tomato, an iconic vegetable in several parts of the world, including Italy (Wang and Liu, 2021). Specifically, relevant phenotypical traits of this crop, such as flowers and fruit, serve as reproductive organs, hence playing a crucial role in real-time monitoring and crop growth assessment (Luo and Li, 2018). For example, HTP, along with machine learning (ML) and deep learning (DL) methods, can help in the assessment and prevention of issues related to shading exposure (Solimani et al., 2023), which can negatively impact fruit quality or the differentiation of ripe and unripe berries.

To this end, DL-based object detectors were successfully applied in the assessment of phenotypical traits in tomato plants (Rong et al., 2023; Cardellicchio et al., 2023). This was related to the development of two categories of detectors: two-stage detectors, such as R-CNN and its successors (Girshick, 2015), and single-stage detectors, whose main exponent is the You Only Look Once (YOLO) family (Redmon

* Corresponding author.
E-mail address: angelo.cardellicchio@stiima.cnr.it (A. Cardellicchio).

https://doi.org/10.1016/j.compag.2024.108728
Received 29 September 2023; Received in revised form 19 January 2024; Accepted 6 February 2024
Available online 9 February 2024

et al., 2016). While both types of detectors provided relevant results in phenotypical traits detection, the algorithms belonging to the YOLO family gathered the focus of the research community due to their practical applications in agriculture (Boogaard et al., 2020; Mahaur and Mishra, 2023; Tian et al., 2023; Zhang et al., 2023). In other words, the models derived by the YOLO family achieved an optimal trade-off between model complexity and real-time performance, allowing the capture of complex patterns and relationships while being deployable on constrained hardware, such as the one available on terrestrial drones and Unmanned Aerial Vehicles (UAVs). Furthermore, researchers focused their efforts on improving the performance of the model, using advanced techniques such as hyper-parameters optimization and data augmentation (Mu et al., 2020; Magalhães et al., 2021; Lawal, 2021). Still, these methods have to deal with many issues related to the phenomena under analysis: one of the most relevant is *data imbalance*, which implies an uneven distribution of samples among different classes and is likely to occur in real-world scenarios, where, for example, it is expected to have a much higher density of some specific phenotyping traits (e.g., nodes) over others (e.g., fruit and flowers). Furthermore, the traits of interest are often of small size. Hence, the use of the bare YOLO architecture may provide suboptimal results due to a lack of optimization for the task at hand.

This work proposes a framework for phenotypical trait detection that deals with the aforementioned issues. The framework is based on the latest iteration of the YOLO family, YOLOv8, and proposes two main points.

- First, a data balancing and augmentation strategy to effectively improve the overall detection performance while retaining the effectiveness and deployability of the model under constrained conditions is proposed. Specifically, this strategy involves a data generation step, where data are artificially generated by applying relevant image processing (IP) techniques, and a balancing step, where the classes are further balanced. These steps resulted in an improved prediction accuracy.
- Second, the architecture of YOLOv8 is improved by adding an attention module, therefore improving the efficiency of the network in dealing with objects of small size.

An extensive comparative evaluation between the approach previously used in Cardellicchio et al. (2023) is then performed, and the results show that the proposed methodology provides improved results, ensuring the reliable detection of tomato plants.

2. Related work

Many studies have focused in recent years on identifying and classifying tomato traits using models based on the YOLO family, as described in Table 1.

Earlier works were based on the third iteration of the YOLO family, that is, YOLOv3. For example, authors in Liu et al. (2020) introduced a modified version of YOLOv3, called YOLO-Tomato, specifically designed to deal with lighting changes, overlapping, and occlusions. The proposed model used a denser architecture and circular bounding boxes, achieving an overall *mean average precision* (mAP) of 94.58% under challenging conditions using a dataset of 609 images. The authors provided an improved version of this model in Wang and Liu (2021) by incorporating dense backbone connections and leveraging K-means clustering to improve the box size calculation and facilitate multi-scale training. These modifications resulted in an improvement in terms of mAP, achieving an overall value of 96.41%. In a separate investigation, Lawal (2021) proposed two distinct models derived from YOLOv3 by replacing the existing backbone either with DenseNet (YOLO-DenseNet) or with a combination of the existing architecture and DenseNet (YOLO-MixNet). YOLO-MixNet achieved the best performance in the tests, with an mAP of 98.40% on the proposed dataset.

Table 1
Results achieved by models based on the YOLO family in tomato traits identification and classification.

Reference	Phenotypical traits	Images (#)	mAP (%)
Liu et al. (2020)	Fruit	966	94.58%
Lawal (2021)	Fruit	425	98.40%
Wang and Liu (2021)	Fruit	3165	96.41%
Ruparelia et al. (2022)	Fruit	2000	81.28%
Zheng et al. (2022)	Fruit	1698	94.44%
Qi et al. (2022)	Fruit	1036	94.10%
Cardellicchio et al. (2023)	Fruit Flowers Nodes	1683	67.90%
Zeng et al. (2023)	Fruit	932	96.90%
Li et al. (2023)	Fruit	6000	97.42%
Mbouembe et al. (2023)	Fruit	966	98.50%
Rong et al. (2023)	Fruit	574	74.80%
Wang et al. (2023b)	Fruit	230	98.80%
Zhang et al. (2023)	Fruit, Flowers	946	86.16%
Yang et al. (2023)	Fruit	922	93.40%

Over time, YOLOv3 was superseded by YOLOv4, which achieved improved performance, as shown by Ruparelia et al. (2022), where the authors compared YOLOv3 and YOLOv4 over a dataset composed of 2000 images. The research demonstrated that $YOLOv4$ was able to exhibit a mAP of 81.28%, while YOLOv3 achieved a lower mAP of 78.49%, highlighting the advantage of denser models in achieving enhanced precision in phenotypic traits detection. Networks derived from the standard YOLOv4 but including some modifications, mainly in the backbone, also showed interesting results. For example, Zheng et al. (2022) achieved a mAP of 94.44% after the introduction of several changes in the original CSPDarkNet53 backbone against a dataset comprising 1698 images featuring tomatoes at various stages of maturity. Another example was the work proposed by Roy and Bhaduri (2022), where the authors customized the backbone to improve the receptive field and preserve accurate localized information, achieving an mAP of 96.29% against a dataset comprising about 12 000 images of tomatoes affected by four distinct plant diseases. In Mbouembe et al. (2023), the authors focused on the neck, replacing the CSP modules with a lightweight version and the standard neck sampling operator with *content-aware reassembly of features* (CARAFE), achieving an mAP of 82.8% on the proposed dataset.

Introducing the denser YOLOv5 model allowed further improvement in localization and detection results. Specifically, Cardellicchio et al. (2023) conducted an investigation on the capabilities of different versions of the YOLOv5 base model, exploring its effectiveness in the identification of three phenotypic traits, that is, flowers, fruits, and nodes, on a challenging dataset. Interestingly, the research demonstrated that the denser YOLOv5 models were able to both reduce false negatives and correctly label objects that were missed on purpose during the labeling step. Other authors also proposed several enhancements to the original model. For example, Qi et al. (2022) modified the standard backbone via Squeeze-and-Excitation (SE) modules, achieving a 94.10% on the proposed dataset. Another proposal was made by Rong et al. (2023) with their YOLOv5-4D model, which combined object detection, multiple object tracking, and specific tracking area counting to effectively count tomato clusters, achieving an mAP of 74.8% on the proposed dataset. The authors in Li et al. (2023) modified the YOLOv5s standard model, introducing a stepwise partial network to enhance the inference speed of the network, and replaced the complete loss of Intersection over union (CIoU) with the efficient loss of Intersection over union (EIoU) to optimize the prediction box regression process. These changes improved the mAP of the original YOLOv5s model of 0.66% on the proposed dataset. To further reduce the computational cost associated with the development of denser models, the authors in Zeng et al. (2023) proposed THYOLO, an algorithm aimed at reducing the

computational cost by combining channel pruning and the optimization of key hyper-parameters, achieving an overall reduction of parameters of 84.15%, while keeping comparable performances. Another approach was proposed by SM-YOLOv5, developed by Wang et al. (2023b), which replaced the original backbone with the MobileNetV3-Large network. This reduced both the computational cost and the model weight, making it deployable easily on constrained robots. The model also achieved interesting results, with an mAP of 98.80% on the proposed dataset.

To address the identification of small targets, such as flowers and tomato fruits, Zhang et al. (2023) proposed a variant to the standard YOLO architecture using a detachable head, thus removing the requirement of pre-determined anchor boxes. The authors used two variants of this network, namely YOLOXMOB and YOLOXPC, which achieved an mAP of 62.10% and 77.33%, respectively, surpassing the value achieved by the bare network on the proposed dataset. Finally, Yang et al. (2023) proposed an enhancement to the YOLOv8 architecture specifically tailored for tomato harvesting automation, implementing a feature enhancement module to improve feature extraction, replacing deeply separable convolution with regular convolution to reduce computational complexity, and introducing a two-way attention gate for enhancing the overall recognition accuracy. These modifications lead to an overall mAP of 93.4% on the proposed dataset, reducing the overall number of parameters required.

As it can be seen from the literature review, at the time of writing, there was a minimal focus on the implication of the use of the YOLOv8 model in plant phenotyping, leaving questions regarding its effectiveness on a complex tomato dataset and the potential enhancement of its architecture unanswered. As such, this work aims to bridge this knowledge gap by investigating the advantages offered by architectural refinements, ultimately contributing to the more precise identification of fruits, flowers, and nodes within the domain of plant phenotyping.

3. Materials and methods

In this section, the dataset used for the analysis will be first described in Section 3.1. Then, the balancing of the classes is discussed in Section 3.2. Subsequently, in Section 3.3, the YOLOv8 model is reviewed. Finally, a brief overview of the criteria used to improve the base YOLOv8 network is provided in Section 3.4.

3.1. Dataset description

The dataset used in this work was composed of 1673 images, each captured at a standardized resolution of 1624 × 1234 pixels. These data, already used in Cardellicchio et al. (2023), were gathered at the HTP platform located at the ALSIA Metapontum Agrobios Research Centre. The images in the dataset encompass three categories of objects related to phenotypic traits: flowers, fruits, and nodes. It is important to underline that the intrinsic structure of the tomato plant makes these traits hard to recognize. For example, some branches may have clusters of large tomatoes, while others may present smaller ones. Additionally, some fruits may be positioned close to the nodes, leading to overlaps between the nodes and the fruit, making it challenging for the model to provide an accurate analysis. These complexities are further exacerbated when encountering data imbalance in real-world scenarios. It is important to highlight that the labeled dataset demonstrates an imbalanced distribution of classes. Specifically, the node class is more than double samples if compared to the other two classes, as depicted in Table 2. This skewed distribution between nodes, flowers, and fruits has the potential to impact the ability of the model to classify instances correctly. Indeed, there is a risk that the model may incorrectly classify nodes as fruits due to this imbalance, primarily stemming from the similarity in color between nodes and unripe fruits, both predominantly displaying a green hue. To mitigate this issue, one strategy is to balance the class distribution, which could contribute to alleviating this challenge to some extent. Furthermore, as fruits grow, they transition to a yellow hue. Consequently, when they are smaller, they may be similar to the appearance of flowers (as they are yellow). Therefore, this can be another risk of the model misclassifying between flowers and fruits.

Table 2
Number of instances per class before and after data balancing.

Class	Before balancing	After balancing
Fruit	1862	4614
Nodes	9276	4744
Flowers	3111	4925

3.2. Data balance

The first challenge to address in this work is the class imbalance among different categories within the dataset. Imbalanced datasets pose significant challenges for data-driven algorithms, primarily due to the unequal distribution of samples across different classes. This imbalance can introduce biases and hinder the ability of the algorithm to comprehend and effectively learn from under-represented classes (Ruiz-Ponce et al., 2023). Consequently, the generalization capabilities of the algorithm are compromised, leading to potential inaccuracies and limited performance when applied to real-world scenarios. Various techniques are available to address data imbalances. In this specific scenario, performing a naive random image-based under-sampling is not a viable option, as it may operate on images instead of labels, potentially causing the inadvertent removal of objects belonging to minority classes instead of the ones belonging to the majority classes. A potential improvement may be disregarding a certain percentage of objects belonging to the majority class. However, this could result in an uptick in false positives during the prediction as the model attempts to learn and identify nodes.

In response to these challenges, this paper proposes a way that strategically overlaps the flowers and fruit with existing nodes, effectively concealing the latter from the attention of the model and, consequently, causing it to disregard them during the prediction. In other words, data are "amalgamated" to generate supplementary samples for underrepresented classes. As for the new instances, these were designed to capture the characteristics of the minority classes closely and were created by using various manipulation techniques on existing data.

To adopt this approach, an analysis was conducted to determine the number of instances within the majority class that did not overlap with other classes or, in other words, exhibited a value for the *Intersection over Union* (IoU) equal to 0. A similar calculation was then also performed for the minority classes. This evaluation estimated the number of new samples that should be generated. Then, the extracted objects were combined with the original images from the dataset to obtain new instances. The number of instances per class before and after data balancing is summarized in Table 2.

The following formula can be used to calculate the required number of samples for achieving balance:

$$\frac{N_{\text{fruit}} + N_{\text{flower}} + X}{(N_{\text{node}} - X) + (N_{\text{fruit}} + N_{\text{flower}} + X)} = \frac{2}{3} \tag{1}$$

where:

- N_{fruit} represents the number of fruit class.
- N_{flower} represents the number of flower class.
- N_{node} represents the number of node class.
- X represents the number of samples deducted from N_{node} and added to N_{fruit} and N_{flower}.

Therefore, the proposed method generated new data by introducing fruit and flowers on the coordinate of empty nodes, i.e., nodes without

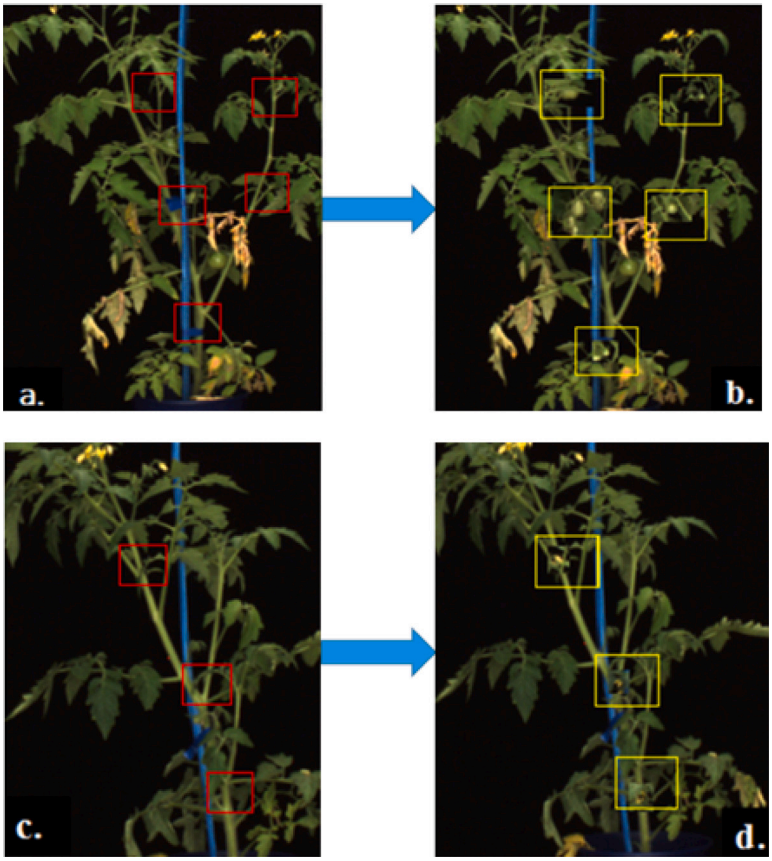

Fig. 1. Comparison between an instance of the dataset before and after the data balancing process. (a) Instance with an empty node; (b) Effect of adding fruits on empty nodes; (c) Instance with an empty node; (d) Effect of adding flowers on empty nodes.

any existing overlap with fruit and flowers. This approach aligns with a similar method proposed by Ruiz-Ponce et al. (2023). In Fig. 1, the comparison between the original instances before and after data balancing is shown.

3.3. YOLOv8 for object detection

You Only Look Once (YOLO) was initially proposed by Redmon et al. (2016), and revolutionized object detection by introducing an end-to-end network capable of simultaneously detecting object bounding boxes and classifying their labels. Since then, YOLO has evolved throughout a series of iterations, reaching the eighth version in January 2023 (Jocher et al., 2023). Specifically, the latest iteration focuses on the following key elements:

- **Backbone**: The YOLOv8 backbone uses a variant of *Cross Partial Stage* (CSP) (Wang et al., 2020), which divides the feature map into separate components for convolution operations and their outputs, resulting in an overall reduced computational complexity, while also retaining the learning capability of the detector. As such, YOLOv8 bases its backbone on the C2f module, a faster implementation of the CSP inspired by the ELAN structure used in YOLOv7 (Wang et al., 2023a). Furthermore, using the SPPF module allows the backbone to improve the detection across different scales.
- **Neck**: The neck of YOLOv8 uses the PAN-FPN module for effective feature fusion across different scales. This module exploits a multi-scale fusion approach using the FPN and PAN architectures, where upper layers capture richer information while lower layers retain specific location details. YOLOv8
- **Head**: YOLOv8 introduces a decoupled head architecture that separates the classification and detection processes. In contrast to the previous anchor-based method, YOLOv8 adopts an anchor-free approach, which locates objects based on their centers, and predicts the distances from them to the bounding box, thus removing the need for predefined anchors.

An overview of the structure of the YOLOv8 model is shown in Fig. 2.

YOLOv8 was the choice for the basic model to be used in this work due to its lightweight architecture, which enables real-time object detection, and its effectiveness at multiple scales. The overall framework used in this work is shown in Fig. 3

3.4. Adding attention to YOLOv8

The Squeeze-and-Excitation (SE) block, as proposed by Hu et al. (2018), is a widely adopted attention mechanism crafted to characterize the relationships among channels. This mechanism enables the network to recalibrate its features effectively, granting it the ability to selectively amplify valuable features by harnessing global information while diminishing the importance of less relevant ones (Lu et al., 2023).

In tomato image analysis, the challenge arises from the similarity in color between nodes, unripe fruits, and the background of the images (which typically consists of leaves). Additionally, the diminutive size of target classes (flowers, fruits, and nodes) exacerbates the difficulty of distinguishing them accurately. Consequently, the model may inadvertently distribute its weights uniformly across the entire image dataset. Accordingly, a strategic approach is needed because a substantial portion of these images lacks utility for our purposes. To refocus the attention of the model on the relevant classes, the idea was

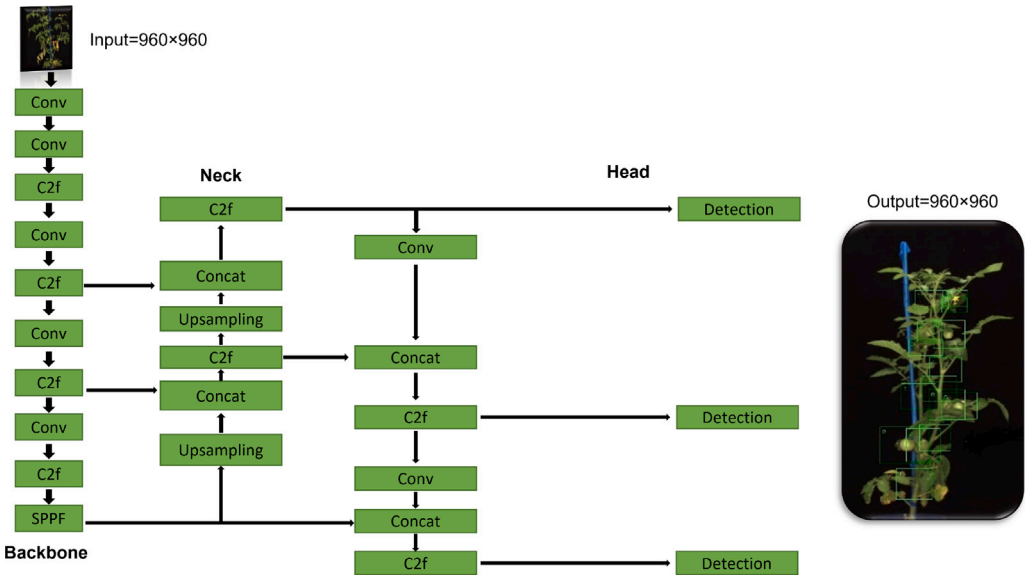

Fig. 2. The architecture of the YOLOv8 model.

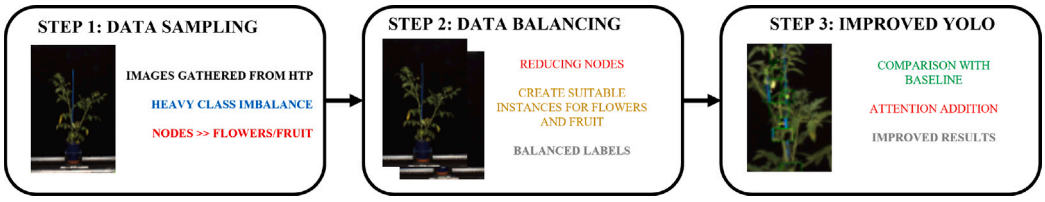

Fig. 3. The processing framework proposed within this work. First, images are gathered directly from a data source like an HTP platform. Then, data augmentation and balancing steps are used to gather a suitable dataset. Finally, several improvements are added to the bare YOLOv8 architecture to improve results.

to embed the SE module as a preprocessing step in the head of the YOLOv8 model.

The placement of the SE-block module, shown in Fig. 4, was determined starting from the considerations provided by the original authors, who highlighted that, even if the optimal location may vary and should be identified through an empirical process, in general, the SE-block achieves suboptimal performance when placed after a *concat* operation (Hu et al., 2018). Hence, the SE-block module was placed after the C2f modules of the original architecture, focusing attention on the features extracted from these layers. The overall architecture of the modified YOLOv8 with the SE-block module is shown in Fig. 5.

3.4.1. Theory behind the Squeeze-and-Excitation module

A SE-block constitutes a computational unit that can be constructed using a transformation F_{tr}, which maps an input $X \in \mathbb{R}^{H' \times W' \times C'}$ to feature maps $U \in \mathbb{R}^{H \times W \times C}$.

Therefore, the output can be written as:

$$y_c = v_c \cdot X = \sum_{s=1}^{C'} v_c^s \cdot x^s \tag{2}$$

Addressing channel dependencies, the focus is on channel-specific signals in output features. However, local receptive fields limit contextual exploitation. Global spatial data were incorporated into channel descriptors using global average pooling to overcome this, producing statistics $Z \in \mathbb{R}^C$.

$$z_c = F_{sq}(y_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} y_c(i,j) \tag{3}$$

Channel dependencies are consequently considered to harness squeezed information. This operation must be flexible, accommodating nonlinear interactions and supporting non-mutually-exclusive relationships. This was achieved using a sigmoid-activated gating mechanism to meet these criteria.

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z)) \tag{4}$$

For model simplicity and enhanced generalization, the SE-block used in this work contains two fully connected layers around the non-linearity.

$$\tilde{x}_c = F_{scale}(y_c, s_c) = s_c \cdot y_c \tag{5}$$

3.4.2. Metrics

Finally, the evaluation metrics used in this work are mainly four, that is, *precision* (P), *recall* (R), F1-score, and *mean average precision* (mAP). Let us briefly recall how precision and recall are computed. For the sake of simplicity, the discussion will be limited to the *binary* case, that is, a classification problem that accounts for two classes, one labeled as *positives*, and the other labeled as *negatives*. The following equations show the formulas for P and R.

$$P = \frac{TP}{TP + FP} \tag{6}$$

$$R = \frac{TP}{TP + FN} \tag{7}$$

In the previous Equations:

- **TP** are the *true positives*, that is, the instances correctly identified by the model as samples of the *positive* class.
- **TN** are the *true negatives*, that is, the instances correctly identified by the model as samples of the *negative* class.
- **FP** are the *false positives*, that is, the instances incorrectly identified by the model as samples of the positive class.

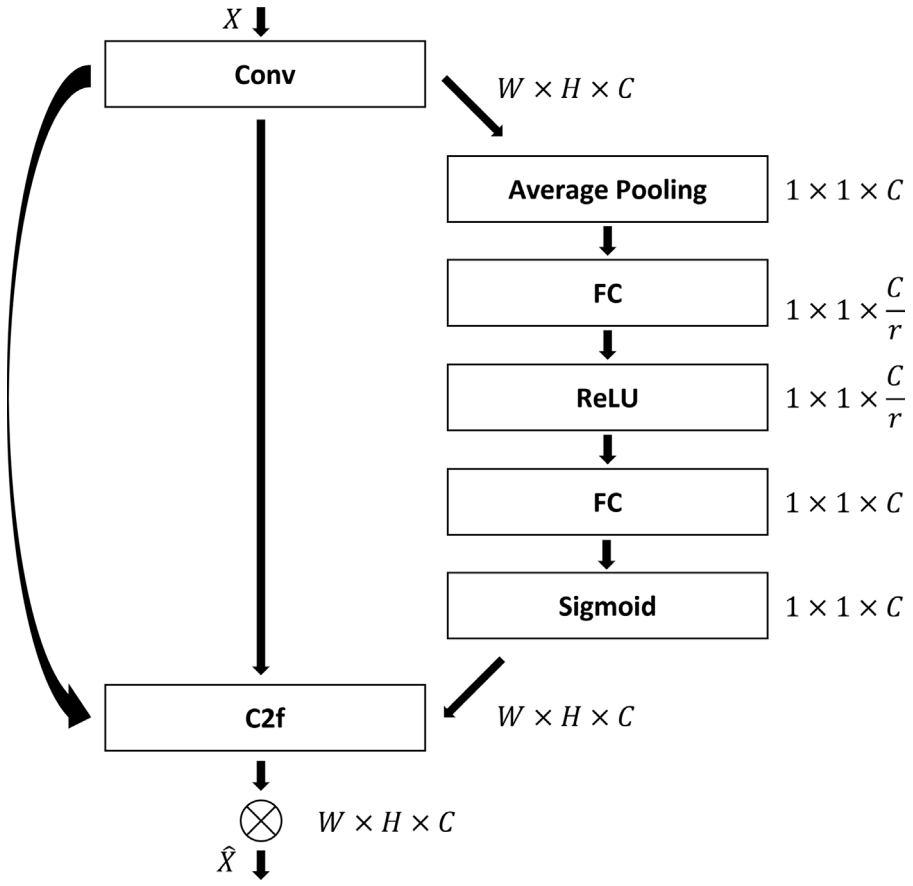

Fig. 4. The result of embedding the SE-block within the C2f and Conv modules.

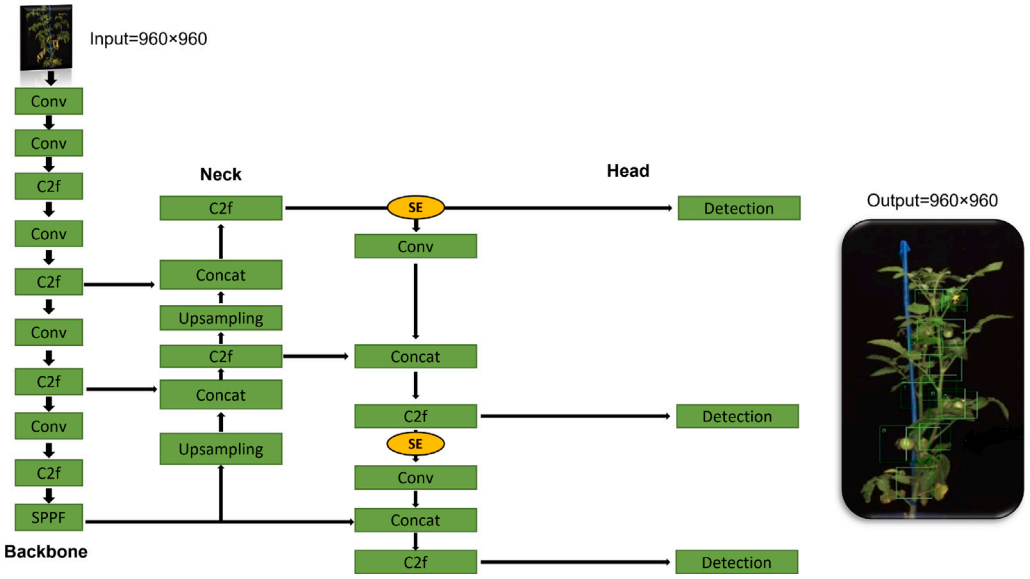

Fig. 5. The proposed architecture, with the addition of the SE-block modules.

- **FN** are the *false negatives*, that is, the instances incorrectly identified by the model as samples of the negative class.

The F1-score can be derived from P and R as follows:

$$F1 = 2\frac{P \cdot R}{P + R} \tag{8}$$

In a multi-class problem, such as the one under investigation, these metrics are computed for all the possible pairs of classes and finally averaged according to the overall number of samples belonging to each class. The evaluation of the mAP requires the introduction of the concept of *Intersection over Union* (IoU), defined as:

$$IoU = \frac{A_O}{A_U} \tag{9}$$

In Eq. (9), A_O is the overlap area between the ground truth and its corresponding predicted box, while A_U is the union between the

Table 3
Training parameters settings.

Parameter	Values
Batch size	4
Image size	960 × 960
Initial learning rate	0.01
Final learning rate	0.001
Weight decay	0.937
Momentum	0.0005

abovementioned areas. IoU can have a value defined within the $[0, 1]$ range and is directly proportional to the overlap between the predicted box and its corresponding ground truth. In practice, when $IoU = 1$, the predicted box completely overlaps the ground truth, while when $IoU = 0$, no pixels of the ground truth are contained within the predicted bounding box.

The $IoU = 0.5$ is the most commonly used threshold to confirm the detection. Starting from this value, it is possible to compute the *average precision* (AP) as the area under the *precision–recall* curve computed at the given IoU threshold. As this value is computed per class, the mAP is the average value of the AP over all classes. In the performed experiments, two values were evaluated for the mAP:

- $mAP - 0.5$, which is the value for the mAP computed considering an IoU threshold of 0.5.
- $mAP - 0.5 - 0.95$, which is the value for the mAP computed for 10 different IoU threshold ranging from 0.5 to 0.95 at a step frequency of 0.05, and then averaged.

4. Experiments and results

4.1. Experimental setup

The machine used for the experiments was based on a Windows 11 operating system, equipped with an NVIDIA GeForce RTX 3080 GPU with 10 GB of RAM and an Intel Core i9-11900HK CPU with 32 GB of RAM. The framework used for deep learning was based on the Ultralytics package and PyTorch 1.11.0.

As for the dataset, YOLOv8 accepts, by default, images with a fixed size of 640 × 640. On the one hand, this resolution can be chosen to overcome the high requirements in terms of the memory computational load of the network, making it feasible, especially for denser models, to be trained on (relatively) constrained machines. At the same time, it is important to underline that using this resolution may compromise the visual appearance of the objects of interest, as most occupy only small patches of the original image. To capture detailed information while still keeping the computation feasible, the images were fed to the network with a fixed size of 960 × 960.

To train the algorithms, two optimization algorithms were tested, that is, *stochastic gradient descent* (SGD), and Adam. The comparison between these algorithms was motivated by several findings. For example, the authors in Yuan and Gao (2020) state that SGD is fast and has low computational requirements but is strongly susceptible to fixed learning rates, as noted in Ding et al. (2019), Luo et al. (2019), and Wu and Liu (2023). To address the latest issue, authors in Carvalho et al. (2020) suggest adopting a flexible learning rate, reducing it progressively through the training process. As for Adam, it is less susceptible to the learning rate, as shown in Llugsi et al. (2021), and Kunstner et al. (2023) highlights how it yields more accurate gradient estimates, even if, as noted in Wilson et al. (2017), its generalization capabilities can be less effective if compared with SGD.

A fixed set of parameters was experimentally set to provide a fair comparison between the algorithms, as shown in Table 3. It is important to underline how the adaptive learning rate was used, as the parameter was gradually reduced from 0.01 to 0.001 during training.

Table 4
Results of the comparison between YOLOv8n and Fast R-CNN on imbalanced data.

Model	mAP0.5	mAP0.95
YOLOv8n	**65.08**%	**19.12**%
Fast R-CNN	26.29%	7.50%

4.2. Comparison with two-stages detectors

To validate the effectiveness of the proposed approach, a comparison with another state-of-the-art method, Faster R-CNN, was performed. Specifically, the Faster R-CNN model was proposed by Girshick (2015) as a two-stage detector, embedding a first step where a region proposal network proposes suitable areas of the image for object localization, and a second step, where a classification model establishes the most likely class to which the objects within the proposed area belong.

As such, the baseline YOLOv8n model and Fast R-CNN were compared. To ensure the fairness of the comparison, the models were trained on the imbalanced dataset for 100 epochs. This was specifically designed to evaluate whether the bare YOLOv8 model had more representational capability of one of the most performing two-stage detectors. The results are shown in terms of mAP in Table 4.

Results show that the baseline model vastly outperforms Fast R-CNN on the proposed dataset. This is also confirmed by looking at some of the predictions performed by Fast R-CNN, shown in Fig. 6. The network could partially identify objects belonging to the majority class (i.e., nodes) but underperforms in identifying traits belonging to the two minority classes. As such, it is safe to assume that, in the specific context, YOLOv8 outperforms Fast R-CNN, and should be selected as the base architecture for object identification.

4.3. Evaluation of the impact of data augmentation

The effectiveness of data augmentation was evaluated using the base YOLOv8n model. Specifically, four transforms were tested, that is:

- **HSV**, where new images were generated with an increment V value to enhance the differences between fruits and nodes.
- **Translate**, where new images were generated by translating different image patches.
- **Scale**, where new images were generated by scaling the original ones.
- **Flip**, where new instances were generated by randomly flipping the original data vertically or horizontally.

The results, shown in Table 5, demonstrate slight improvements in the mAP scores, especially on minority classes with the HSV, Translate, and Scale operations. The most impactful adjustment was achieved when the Scale transform was considered, resulting in the best mAP performance on both minority classes. As such, it can be safely assumed that introducing data augmentation enhances the sensitivity of the model by allowing it to focus on small objects, which is particularly beneficial given the specificities of the dataset used in this work.

4.4. Comparison of the SGD and adam optimizers

In this section, the performance of both the optimizers are compared. In this case, the comparison was performed using all the densities provided by YOLOv8, ranging from the sparser model (YOLOv8n) to the denser one (YOLOv8x). It is important to underline that this evaluation was performed on the base dataset (i.e., the dataset without augmentation). This was experimentally chosen to assess the effectiveness of the compared algorithms fairly. The results are shown in Table 6.

The results in Table 6 highlight that SGD provides a better outcome regarding all metrics. Moreover, the best-performing model is

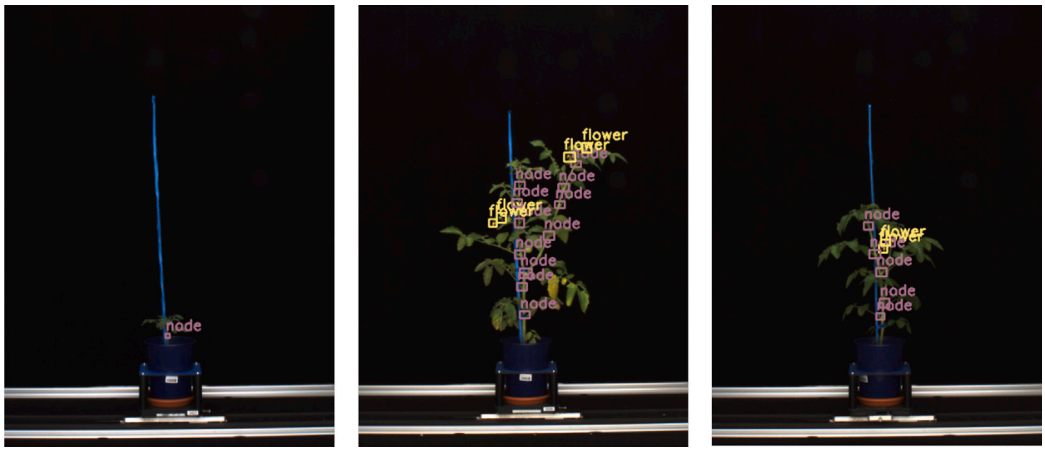

Fig. 6. Predictions performed by Fast R-CNN.

Table 5
Comparison of the results provided by using different data augmentation methods on YOLOv8n with the proposed approach.

Hyper-parameter	Class	mAP50%	mAP50 – 95%
Baseline	Fruit	66.20%	19.26%
	Flower	57.31%	18.34%
	Node	**65.08%**	19.12%
HSV (V)	Fruit	**68.30%**	19.44%
	Flower	59.07%	18.34%
	Node	63.03%	19.02%
Translate	Fruit	67.25%	**20.49%**
	Flower	**64.35%**	19.17%
	Node	57.02%	**20.11%**
Flip	Fruit	66.37%	20.14%
	Flower	61.24%	19.27%
	Node	64.02%	20.01%
Scale	Fruit	**68.30%**	20.14%
	Flower	61.14%	**20.10%**
	Node	62.07%	20.04%

Table 6
Results of evaluating different data augmentation methods on YOLOv8n with the proposed approach.

Optimizer	Models	P%	R%	F1	mAP50%	mAP50%–95%
SGD	YOLOv8n	58.63%	58.98%	58.80%	53.69%	18.75%
	YOLOv8s	62.56%	62.90%	62.72%	58.25%	20.56%
	YOLOv8m	63.93%	61.35%	62.61%	58.14%	20.53%
	YOLOv8l	62.49%	62.26%	62.37%	56.88%	20.22%
	YOLOv8x	**69.78%**	**63.12%**	**65.28%**	**64.09%**	**23.15%**
Adam	YOLOv8n	57.08%	57.12%	57.09%	52.70%	18.10%
	YOLOv8s	61.36%	59.61%	60.47%	55.31%	19.63%
	YOLOv8m	60.43%	61.19%	60.80%	56.19%	19.70%
	YOLOv8l	61.16%	60.44%	60.79%	55.07%	18.68%
	YOLOv8x	58.79%	59.66%	59.22%	53.98%	18.86%

YOLOv8x, which shows a noticeable difference in precision (about 10%) and mAP50 (about 11%). As such, it can be safely assumed that SGD outperforms Adam. This can be explained as follows. In the scenario under analysis, an imbalanced dataset where the minority classes had a noticeably smaller number of samples if compared with the majority class, the dynamic adjustment of the learning rate used by SGD was probably able to play a crucial role in effectively guiding the optimization process. Conversely, the adaptive learning rate algorithm used by Adam was probably less suited to this particular scenario. Moreover, the simplicity of the SGD update rule likely contributed to its capability of preventing over-fitting on the limited data available for the minority classes. In other words, by incorporating class weights during training, SGD implemented a prioritized learning approach that assigns higher weights to the instances belonging to the minority classes, resulting in observed performance improvements.

Consequently, the decision was made to proceed with the rest of the tests using SGD optimizer. Still, an important aspect that must still be addressed is the impact of balancing the data distribution on the outcome of the analysis.

Table 7
Results of the comparison between imbalanced and balanced data.

Dataset	Models	P%	R%	F1	mAP50%	mAP50%–95%
Imbalance	YOLOv8n	58.63%	58.98%	58.80%	53.69%	18.75%
	YOLOv8s	62.56%	62.90%	62.72%	58.25%	20.56%
	YOLOv8m	63.93%	61.35%	62.61%	58.14%	20.53%
	YOLOv8l	62.49%	62.26%	62.37%	56.88%	20.22%
	YOLOv8x	69.78%	**63.12%**	**66.28%**	**64.09%**	**23.15%**
Balance	YOLOv8n	64.65%	58.35%	61.33%	59.77%	21.42%
	YOLOv8s	66.81%	60.55%	63.52%	61.40%	22.22%
	YOLOv8m	69.32%	60.31%	64.50%	61.94%	22.69%
	YOLOv8l	**70.84%**	60.73%	65.39%	62.11%	22.44%
	YOLOv8x	69.37%	59.52%	64.06%	61.66%	22.12%

4.5. Evaluation of the impact of data balancing

The impact of data balancing was evaluated by comparing the performance of different models on imbalanced and balanced data. The results are shown in Table 7.

From the results presented in Table 7, the impact of data balancing is clear, with an increment between 3% and 8% for all metrics on almost all the proposed models.

Let us consider briefly the results achieved. As already shown in Section 4.4, YOLOv8x is the model that is able to achieve the best results, probably thanks to the large capacity of the model. However, the results are biased towards the majority class; consequently, the model would probably present reduced generalization capabilities. When data are balanced, however, smaller models are able to achieve performance comparable to the ones from YOLOv8x, mainly due to the availability of an adequate amount of samples that properly characterize the data generation mechanism. Consequently, when balanced data are considered, YOLOv8l achieves the best performance. Interestingly, when data are balanced, YOLOv8x appears to be affected by the double descent phenomena (Nakkiran et al., 2021), hence its performance decrease.

4.6. Embedding the SE-block attention module

After applying data balancing, let us evaluate the effects of embedding the SE-block attention module, as described in Section 3.4. The results are described in Table 8.

Table 8
Results of evaluating balanced data using the attention mechanism.

Model	P%	R%	F1	mAP50%	mAP50%–95%
YOLOv8n	66.08%	52.23%	58.34%	55.85%	18.77%
YOLOv8s	66.37%	58.40%	62.13%	60.47%	21.23%
YOLOv8m	68.49%	58.74%	63.24%	60.03%	20.49%
YOLOv8l	68.52%	**60.68%**	**64.36%**	60.52%	21.33%
YOLOv8x	**69.23%**	56.87%	62.44%	**60.57%**	**21.94%**

Table 9
Results of evaluating imbalanced data using the attention mechanism.

Model	*P%*	*R%*	F1	mAP50%	mAP50%–95%
YOLOv8n	69.48%	57.31%	62.81%	60.85%	20.90%
YOLOv8s	68.66%	62.51%	65.44%	64.25%	22.98%
YOLOv8m	**71.03%**	**63.66%**	**67.14%**	**65.82%**	**23.70%**
YOLOv8l	70.50%	63.31%	66.71%	64.62%	22.61%
YOLOv8x	69.60%	64.01%	66.68%	64.67%	22.99%

Table 10
Results of evaluating imbalanced data using the attention mechanism and pre-trained weights obtained from the balanced dataset.

Models	P%	R%	F1	mAP50%	mAP50%–95%
YOLOv8n	70.01%	60.69%	65.01%	63.69%	22.33%
YOLOv8s	70.20%	64.61%	67.28%	65.65%	23.27%
YOLOv8m	**71.59%**	**64.96%**	**68.11%**	**65.77%**	**23.38%**
YOLOv8l	70.11%	65.59%	67.77%	64.82%	22.61%
YOLOv8x	71.17%	62.15%	66.35%	64.83%	23.89%

Interestingly, there is a decrement in performance across all versions of YOLOv8 when the attention module is applied to the balanced dataset. This effect can be explained by looking at the data balancing process itself, as one of its drawbacks is that, regardless of the augmentation technique used, there is a risk of information loss. In other words, the proposed technique may duplicate existing objects multiple times to achieve data balancing. Hence, some underlying data generation mechanisms may assume a more relevant weight. As the attention mechanisms focus on *local* information, which may be biased by the augmentation process, a decrease in performance can be experienced. Furthermore, as already seen in Section 4.5, the reduced information about the nodes could further impact the overall effectiveness of the model.

As such, the performance of the modified version of YOLOv8 should also be evaluated on imbalanced data. The results of this evaluation are presented in Table 9.

It becomes evident that the models yield improved results on imbalanced data when the attention module is used. Again, this is due to the fact that imbalanced data retains a broader spectrum of available information, which, in this particular situation, may be able to provide better results if compared with data balanced with the method previously proposed.

Still, an alternative approach was followed to deal with the challenge of information loss while still taking advantage of data balancing. Specifically, rather than using balanced data directly, the idea was to train the model on the original, imbalanced data and assign a higher weight to instances belonging to minority classes, effectively achieving a balance between data classes while retaining the original information. This was effectively implemented using a "transfer learning-alike" approach by applying the weights of the model trained on balanced data when dealing with imbalanced data. This approach achieved a noticeable improvement in the results, even if compared with the results achieved on imbalanced data, as described in Table 10.

4.7. Comparing YOLOv5 and YOLOv8

The final comparison proposed in this paper assesses the results achieved by this work with respect to the approach proposed in Cardellicchio et al. (2023). To this end, let us recall that Cardellicchio et al. (2023) evaluated the use of YOLOv5 on the same dataset used in this work; however, it is important to underline that the assessment was performed exclusively on imbalanced data and that no attention mechanisms were placed on the head of the model. Hence, this comparison could help to evaluate the impact of the proposed balancing and attention mechanisms. To compare the results achieved by the two approaches, only the denser architectures (that is, YOLOv5x and YOLOv8x) were considered. The differences between the two approaches were evaluated in terms of precision, recall, and F1 score.

Table 11
Comparison between the results achieved by YOLOv8 and YOLOv5 in Cardellicchio et al. (2023).

Class	Density	YOLOv8			YOLOv5		
		TP	B-FN	B-FP	TP	B-FN	B-FP
Fruit	Small	75.97%	13.75%	20.89%	64.55%	35.71%	70.69%
	Medium	**79.24%**	**11.75%**	18.36%	73.36%	24.57%	70.29%
	Large	79.04%	12.28%	**18.22%**	76.17%	22.30%	70.69%
	eXtra	78.77%	16.36%	19.49%	77.97%	19.83%	80.84%
Nodes	Small	56.26%	47.69%	44.56%	50.45%	48.99%	48.40%
	Medium	61.49%	52.16%	**39.14%**	64.84%	54.00%	54.89%
	Large	60.05%	57.84%	40.66%	**66.66%**	58.83%	58.55%
	eXtra	55.04%	**46.92%**	45.67%	68.71%	59.64%	59.25%
Flowers	Small	**68.02%**	40.02%	**31.86%**	48.41%	51.59%	45.84%
	Medium	67.38%	37.28%	31.90%	56.81%	44.07%	39.49%
	Large	67.34%	**31.50%**	32.02%	64.12%	34.35%	34.71%
	eXtra	65.09%	38.85%	33.67%	64.52%	34.39%	33.15%

Let us start with the precision shown in Fig. 7. The two networks achieve comparable results, even if YOLOv5x presented a sudden drop in precision at around 0.7 confidence, while YOLOv8x shows consistently better results when balancing and attention mechanisms are used. Interestingly, a decline in precision still appeared at a confidence score of 0.8, mainly due to a decline for the node class. This may be interpreted as an intrinsic limitation of the model, which should be addressed in future works.

The recall confirms these findings, shown in Fig. 8, which is consistently higher for YOLOv8x.

Let us also check the values for the F1 score, depicted in Fig. 9. As for YOLOv5, the maximum F1 score was achieved at a confidence score of 0.4. In contrast, YOLOv8 obtained a consistent F1 score, peaking at a confidence level of around 0.6.

This comparison confirms that YOLOv8 is able to characterize objects of interest with a higher confidence score. This is related to the improvements the network added, which shows better representation capabilities, and the introduction of attention mechanisms, which allows for properly characterizing small patches of interest.

Finally, let us delve deeper into the results achieved by both approaches, as shown in Table 11. Here, B-FP corresponds to *background false positives*, that is, boxes detected by the model that lack corresponding labels provided by domain experts. Meanwhile, B-FN represents background false negatives, which refers to labeled bounding boxes not detected by the network.

Although the B-FN occurrence in the YOLOv8 model was relatively lower than YOLOv5, both nodes and flowers still show a high level of B-FN. This indicates that although data balancing in YOLOv8 has helped mitigate this issue to some extent, as noted in the previous study, the primary concern likely lies in the inability of the model to properly characterize the visual appearance of the objects of these classes. This is also true for B-FPs, which highlights the inability of the model to effectively distinguish between primary and secondary nodes of the stems or even from nodes and visually overlapped leaves. Still, the comparison shows that YOLOv8, due to the innovation proposed, consistently outperforms YOLOv5 in terms of all the provided metrics, independently from the considered density.

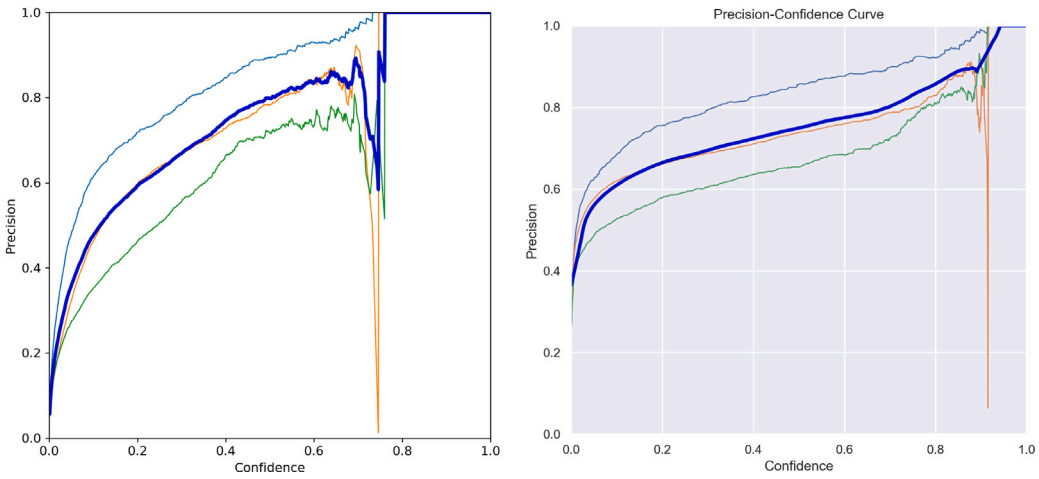

Fig. 7. The precision achieved by the YOLOv5x model (on the left) and the YOLOv8x model (on the right) after applying data balancing and attention. Light blue results are for fruit, orange for nodes, and green for flowers.

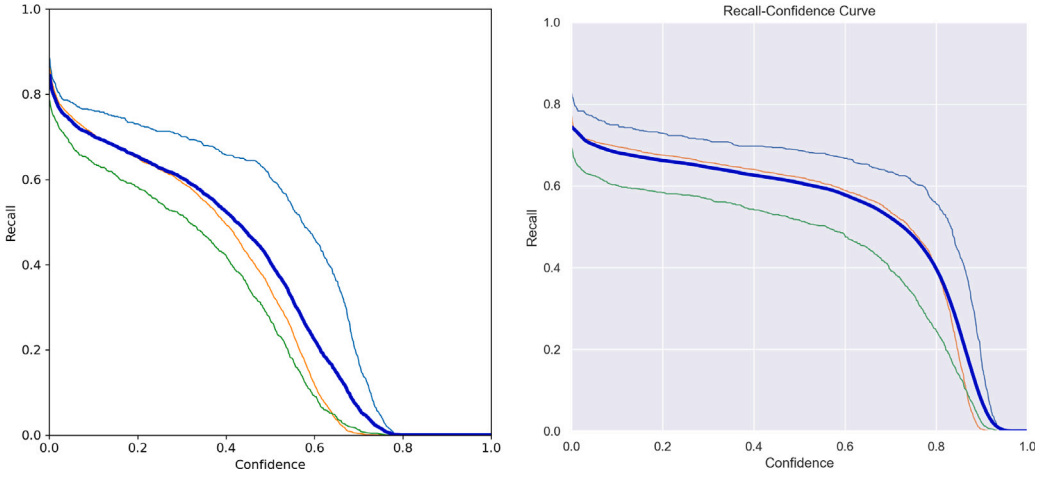

Fig. 8. The recall achieved by the YOLOv5x model (on the left) and the YOLOv8x model (on the right) after applying data balancing and attention. Light blue results are for fruit, orange for nodes, and green for flowers.

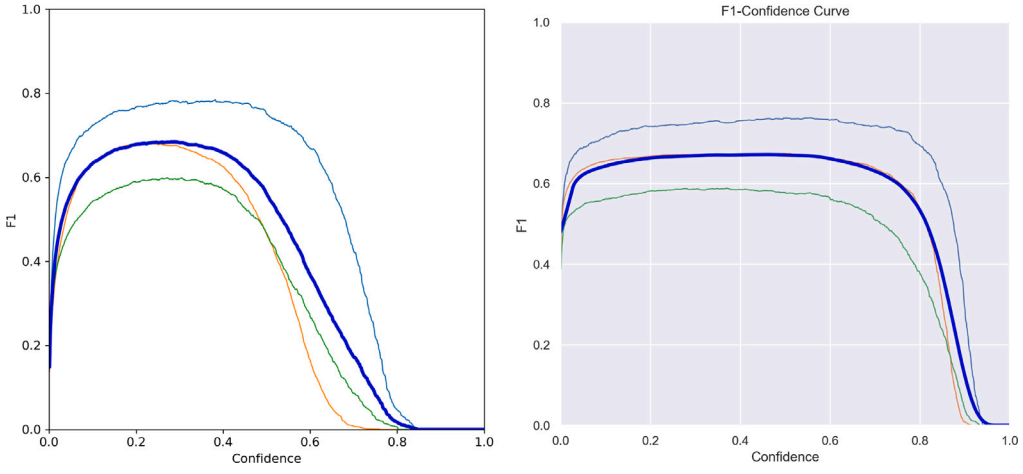

Fig. 9. The F1-score achieved by the YOLOv5x model (on the left) and the YOLOv8x model (on the right) after applying data balancing and attention. Light blue results are for fruit, orange for nodes, and green for flowers.

5. Conclusions and future works

Identifying relevant phenotypical traits in tomato plants is demanding, requiring solutions to enhance data balance for accurate diagnosis. This motivated the proposal of an end-to-end pipeline for data balancing and phenotypical trait detection under challenging conditions using single-stage detectors.

Although this work was mainly focused on challenges centered on a specific tomato plants dataset, the approach is straightforward to adapt to other similar scenarios with low effort. Results showed that by incorporating attention mechanisms, along with a transfer-learning-like method to use best weights achieved on balanced data to evaluate imbalanced data, the accuracy in the detection of relevant phenotypical traits improved significantly.

Still, several limitations remain to be addressed, mainly due to the requirement for semantic information to be embedded within the framework to let the network differentiate, for example, between nodes on primary stems, which are phenotypically relevant, and nodes on secondary stems. As such, future research will focus on this kind of integration, exploiting the knowledge achievable by using other approaches, such as graph neural networks, and exploring other mechanisms to enhance the proposed pipeline further, expanding the experiments to sibling domains.

CRediT authorship contribution statement

Firozeh Solimani: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Writing – original draft. **Angelo Cardellicchio:** Data curation, Formal analysis, Investigation, Methodology, Software, Supervision, Validation, Visualization, Writing – review & editing. **Giovanni Dimauro:** Formal analysis, Supervision, Validation, Writing – review & editing. **Angelo Petrozza:** Resources, Supervision, Writing – review & editing. **Stephan Summerer:** Data curation, Resources, Writing – review & editing. **Francesco Cellini:** Funding acquisition, Project administration, Resources, Writing – review & editing. **Vito Renò:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – review & editing.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgement

The activities described in this work are within the research projects *PHENO – Accordo di collaborazione tra ALSIA e CNR STIIMA – ref. prot. CNR STIIMA 3621/2020*.

References

Afonso, M., Fonteijn, H., Fiorentin, F.S., Lensink, D., Mooij, M., Faber, N., Polder, G., Wehrens, R., 2020. Tomato fruit detection and counting in greenhouses using deep learning. Front. Plant Sci. 11, URL https://www.frontiersin.org/articles/10.3389/fpls.2020.571299.

Bac, C.W., Hemming, J., van Tuijl, B., Barth, R., Wais, E., van Henten, E.J., 2017. Performance evaluation of a harvesting robot for sweet pepper. J. Field Robotics 34 (6), 1123–1139. http://dx.doi.org/10.1002/rob.21709, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21709. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21709.

Boogaard, F.P., Rongen, K.S.A.H., Kootstra, G.W., 2020. Robust node detection and tracking in fruit-vegetable crops using deep learning and multi-view imaging. Biosyst. Eng. 192, 117–132. http://dx.doi.org/10.1016/j.biosystemseng.2020.01.023, URL https://www.sciencedirect.com/science/article/pii/S1537511020300350.

Cardellicchio, A., Solimani, F., Dimauro, G., Petrozza, A., Summerer, S., Cellini, F., Renò, V., 2023. Detection of tomato plant phenotyping traits using YOLOv5-based single stage detectors. Comput. Electron. Agric. 207, 107757. http://dx.doi.org/10.1016/j.compag.2023.107757, URL https://www.sciencedirect.com/science/article/pii/S016816992300145X.

Carvalho, P., Lourenço, N., Assunção, F., Machado, P., 2020. AutoLR: an evolutionary approach to learning rate policies. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference. GECCO '20, Association for Computing Machinery, New York, NY, USA, pp. 672–680. http://dx.doi.org/10.1145/3377930.3390158, URL https://dl.acm.org/doi/10.1145/3377930.3390158.

Ding, J., Ren, X., Luo, R., Sun, X., 2019. An adaptive and momental bound method for stochastic learning. http://dx.doi.org/10.48550/arXiv.1910.12249, URL http://arxiv.org/abs/1910.12249. arXiv:1910.12249 [cs, stat].

Girshick, R., 2015. Fast R-CNN. pp. 1440–1448, URL https://openaccess.thecvf.com/content_iccv_2015/html/Girshick_Fast_R-CNN_ICCV_2015_paper.html.

Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-Excitation Networks. pp. 7132–7141, URL https://openaccess.thecvf.com/content_cvpr_2018/html/Hu_Squeeze-and-Excitation_Networks_CVPR_2018_paper.html.

Jocher, G., Chaurasia, A., Qiu, J., 2023. Ultralytics YOLO. URL https://github.com/ultralytics/ultralytics.

Kunstner, F., Chen, J., Lavington, J.W., Schmidt, M., 2023. Noise is not the main factor behind the gap between SGD and adam on transformers, but sign descent might be. http://dx.doi.org/10.48550/arXiv.2304.13960, URL http://arxiv.org/abs/2304.13960. arXiv:2304.13960 [cs, math].

Lawal, O.M., 2021. Development of tomato detection model for robotic platform using deep learning. Multimedia Tools Appl. 80 (17), 26751–26772. http://dx.doi.org/10.1007/s11042-021-10933-w.

Li, R., Ji, Z., Hu, S., Huang, X., Yang, J., Li, W., 2023. Tomato maturity recognition model based on improved YOLOv5 in greenhouse. Agronomy 13 (2), 603. http://dx.doi.org/10.3390/agronomy13020603, URL https://www.mdpi.com/2073-4395/13/2/603. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

Liu, G., Nouaze, J.C., Touko Mbouembe, P.L., Kim, J.H., 2020. YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3. Sensors 20 (7), 2145. http://dx.doi.org/10.3390/s20072145, URL https://www.mdpi.com/1424-8220/20/7/2145. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.

Llugsi, R., Yacoubi, S.E., Fontaine, A., Lupera, P., 2021. Comparison between adam, AdaMax and adam w optimizers to implement a weather forecast based on neural networks for the andean city of quito. In: 2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM). pp. 1–6. http://dx.doi.org/10.1109/ETCM53643.2021.9590681, URL https://ieeexplore.ieee.org/abstract/document/9590681.

Lu, Z., Miao, J., Dong, J., Zhu, S., Wu, P., Wang, X., Feng, J., 2023. Automatic multilabel classification of multiple fundus diseases based on convolutional neural network with squeeze-and-excitation attention. Transl. Vis. Sci. Technol. 12 (1), 22. http://dx.doi.org/10.1167/tvst.12.1.22.

Luo, H., Li, F., 2018. Tomato yield, quality and water use efficiency under different drip fertigation strategies. Sci. Horticult. 235, 181–188. http://dx.doi.org/10.1016/j.scienta.2018.02.072, URL https://www.sciencedirect.com/science/article/pii/S0304423818301560.

Luo, L., Xiong, Y., Liu, Y., Sun, X., 2019. Adaptive gradient methods with dynamic bound of learning rate. http://dx.doi.org/10.48550/arXiv.1902.09843, URL http://arxiv.org/abs/1902.09843. arXiv:1902.09843 [cs, stat].

Magalhães, S.A., Castro, L., Moreira, G., dos Santos, F.N., Cunha, M., Dias, J., Moreira, A.P., 2021. Evaluating the single-shot multibox detector and YOLO deep learning models for the detection of tomatoes in a greenhouse. Sensors 21 (10), 3569. http://dx.doi.org/10.3390/s21103569, URL https://www.mdpi.com/1424-8220/21/10/3569. Number: 10 Publisher: Multidisciplinary Digital Publishing Institute.

Mahaur, B., Mishra, K.K., 2023. Small-object detection based on YOLOv5 in autonomous driving systems. Pattern Recognit. Lett. 168, 115–122. http://dx.doi.org/10.1016/j.patrec.2023.03.009, URL https://www.sciencedirect.com/science/article/pii/S0167865523000727.

Maji, A.K., Marwaha, S., Kumar, S., Arora, A., Chinnusamy, V., Islam, S., 2022. SlypNet: Spikelet-based yield prediction of wheat using advanced plant phenotyping and computer vision techniques. Front. Plant Sci. 13, URL https://www.frontiersin.org/articles/10.3389/fpls.2022.889853.

Mbouembe, P.L.T., Liu, G., Sikati, J., Kim, S.C., Kim, J.H., 2023. An efficient tomato-detection method based on improved YOLOv4-tiny model in complex environment. Front. Plant Sci. 14, URL https://www.frontiersin.org/articles/10.3389/fpls.2023.1150958.

Mu, Y., Chen, T.-S., Ninomiya, S., Guo, W., 2020. Intact detection of highly occluded immature tomatoes on plants using deep learning techniques. Sensors 20 (10), 2984. http://dx.doi.org/10.3390/s20102984, URL https://www.mdpi.com/1424-8220/20/10/2984. Number: 10 Publisher: Multidisciplinary Digital Publishing Institute.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., Sutskever, I., 2021. Deep double descent: Where bigger models and more data hurt. J. Stat. Mech. Theory Exp. 2021 (12), 124003.

Qi, J., Liu, X., Liu, K., Xu, F., Guo, H., Tian, X., Li, M., Bao, Z., Li, Y., 2022. An improved YOLOv5 model based on visual attention mechanism: Application to recognition of tomato virus disease. Comput. Electron. Agric. 194, 106780. http://dx.doi.org/10.1016/j.compag.2022.106780, URL https://www.sciencedirect.com/science/article/pii/S0168169922000977.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection. pp. 779–788, URL https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html.

Rong, J., Zhou, H., Zhang, F., Yuan, T., Wang, P., 2023. Tomato cluster detection and counting using improved YOLOv5 based on RGB-D fusion. Comput. Electron. Agric. 207, 107741. http://dx.doi.org/10.1016/j.compag.2023.107741, URL https://www.sciencedirect.com/science/article/pii/S0168169923001291.

Roy, A.M., Bhaduri, J., 2022. Real-time growth stage detection model for high degree of occultation using DenseNet-fused YOLOv4. Comput. Electron. Agric. 193, 106694. http://dx.doi.org/10.1016/j.compag.2022.106694, URL https://www.sciencedirect.com/science/article/pii/S0168169922000114.

Ruiz-Ponce, P., Ortiz-Perez, D., Garcia-Rodriguez, J., Kiefer, B., 2023. POSEIDON: A data augmentation tool for small object detection datasets in maritime environments. Sensors 23 (7), 3691. http://dx.doi.org/10.3390/s23073691, URL https://www.mdpi.com/1424-8220/23/7/3691. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.

Ruparelia, S., Jethva, M., Gajjar, R., 2022. Real-time tomato detection, classification, and counting system using deep learning and embedded systems. In: Thakkar, F., Saha, G., Shahnaz, C., Hu, Y.-C. (Eds.), Proceedings of the International E-Conference on Intelligent Systems and Signal Processing. In: Advances in Intelligent Systems and Computing, Springer, Singapore, pp. 511–522. http://dx.doi.org/10.1007/978-981-16-2123-9_39.

Solimani, F., Cardellicchio, A., Nitti, M., Lako, A., Dimauro, G., Renò, V., 2023. A systematic review of effective hardware and software factors affecting high-throughput plant phenotyping. Information 14 (4), 214. http://dx.doi.org/10.3390/info14040214, URL https://www.mdpi.com/2078-2489/14/4/214. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

Tian, Z., Huang, J., Yang, Y., Nie, W., 2023. KCFS-YOLOv5: A high-precision detection method for object detection in aerial remote sensing images. Appl. Sci. 13 (1), 649. http://dx.doi.org/10.3390/app13010649, URL https://www.mdpi.com/2076-3417/13/1/649. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M., 2023a. YOLOv7: Trainable Bag-Of-Freebies Sets New State-Of-The-Art for Real-Time Object Detectors. pp. 7464–7475, URL https://openaccess.thecvf.com/content/CVPR2023/html/Wang_YOLOv7_Trainable_Bag-of-Freebies_Sets_New_State-of-the-Art_for_Real-Time_Object_Detectors_CVPR_2023_paper.html.

Wang, C.-Y., Liao, H.-Y.M., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., Yeh, I.-H., 2020. CSPNet: A new backbone that can enhance learning capability of CNN. pp. 390–391, URL https://openaccess.thecvf.com/content_CVPRW_2020/html/w28/Wang_CSPNet_A_New_Backbone_That_Can_Enhance_Learning_Capability_of_CVPRW_2020_paper.html.

Wang, X., Liu, J., 2021. Tomato anomalies detection in greenhouse scenarios based on YOLO-dense. Front. Plant Sci. 12, URL https://www.frontiersin.org/articles/10.3389/fpls.2021.634103.

Wang, X., Wu, Z., Jia, M., Xu, T., Pan, C., Qi, X., Zhao, M., 2023b. Lightweight SM-YOLOv5 tomato fruit detection algorithm for plant factory. Sensors 23 (6), 3336. http://dx.doi.org/10.3390/s23063336, URL https://www.mdpi.com/1424-8220/23/6/3336. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.

Wilson, A.C., Roelofs, R., Stern, M., Srebro, N., Recht, B., 2017. The marginal value of adaptive gradient methods in machine learning. In: Advances in Neural Information Processing Systems. Vol. 30, Curran Associates, Inc., URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/81b3833e2504647f9d794f7d7b9bf341-Abstract.html.

Wu, Y., Liu, L., 2023. Selecting and composing learning rate policies for deep neural networks. ACM Trans. Intell. Syst. Technol. 14 (2), 22:1–22:25. http://dx.doi.org/10.1145/3570508, URL https://dl.acm.org/doi/10.1145/3570508.

Yang, G., Wang, J., Nie, Z., Yang, H., Yu, S., 2023. A lightweight YOLOv8 tomato detection algorithm combining feature enhancement and attention. Agronomy 13 (7), 1824. http://dx.doi.org/10.3390/agronomy13071824, URL https://www.mdpi.com/2073-4395/13/7/1824. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.

Yuan, W., Gao, K.-X., 2020. EAdam optimizer: How ϵ impact adam. http://dx.doi.org/10.48550/arXiv.2011.02150, URL http://arxiv.org/abs/2011.02150. arXiv:2011.02150 [cs, stat].

Zeng, T., Li, S., Song, Q., Zhong, F., Wei, X., 2023. Lightweight tomato real-time detection method based on improved YOLO and mobile deployment. Comput. Electron. Agric. 205, 107625. http://dx.doi.org/10.1016/j.compag.2023.107625, URL https://www.sciencedirect.com/science/article/pii/S0168169923000133.

Zhang, J., Zhang, J., Zhou, K., Zhang, Y., Chen, H., Yan, X., 2023. An improved YOLOv5-based underwater object-detection framework. Sensors 23 (7), 3693. http://dx.doi.org/10.3390/s23073693, URL https://www.mdpi.com/1424-8220/23/7/3693. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.

Zheng, T., Jiang, M., Li, Y., Feng, M., 2022. Research on tomato detection in natural environment based on RC-YOLOv4. Comput. Electron. Agric. 198, 107029. http://dx.doi.org/10.1016/j.compag.2022.107029, URL https://www.sciencedirect.com/science/article/pii/S0168169922003465.