Deep learning for real-time strawberry detection, ripeness classification, and picking point localization: A review of architectures, field studies, and open challenges

Mohieddine Jelali
TH Köln - University of Applied Sciences
Fabian Gerz
fabian.gerz@th-koeln.de
TH Köln - University of Applied Sciences
Orhan-Timo Altan
TH Köln - University of Applied Sciences
Loui Al-Shrouf
TH Köln - University of Applied Sciences

Research Article

Keywords: Crop Harvesting, Strawberry, Machine Vision, Object Detection, Fruit Detection, Ripeness Classification, Picking Point Localization, Deep Learning Models, YOLO

Posted Date: June 18th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9886956/v1

Additional Declarations: No competing interests reported.

Deep learning for real-time strawberry detection, ripeness classification, and picking point localization: A review of architectures, field studies, and open challenges

Mohieddine Jelali[1*], Fabian Gerz[1*], Orhan-Timo Altan[1] and Loui Al-Shrouf[1]

[1]Cologne Laboratory of Artificial Intelligence and Smart Automation (CAISA), Institute of Product Development and Engineering Design (IPK), Technische Hochschule Köln—University of Applied Sciences, Betzdorfer Str. 2, Cologne, 50679, Germany.

*Corresponding author(s). E-mail(s): mohieddine.jelali@th-koeln.de; fabian.gerz@th-koeln.de;
Contributing authors: timo.altan@th-koeln.de; loui.al-shrouf@th-koeln.de;

Abstract

Accurate detection of strawberry fruit, reliable ripeness estimation, and precise localization of the picking point are essential for automated harvesting and yield prediction in smart farming. However, real-world environments introduce significant challenges, including occlusion, illumination variability, and high visual similarity between ripeness stages. Deep learning (DL)-based object detection methods have become the dominant approach to address these issues.

This paper presents a systematic review of 60 peer-reviewed studies published between 2023 and 2026, focusing on detecting strawberries and their ripeness using the YOLO family (v5–v11) of DL algorithms. The studies are analyzed with respect to dataset characteristics, preprocessing and augmentation strategies, model architectures, and evaluation protocols.

Our results show a clear dominance of YOLOv8, used in 28 (46.7%) of the 60 reviewed works, due to its real-time capability and architectural flexibility. Despite its short history, YOLOv11 has been adopted in 13 studies (21.7%) owing to its balanced precision and computational efficiency. Hybrid CNN–ViT models

that integrate Transformer modules or networks into YOLO are gaining attention (8 studies, 13.3%) and show improved performance in complex scenarios, but they still incur higher computational cost.
However, we identify critical methodological issues that affect the validity of reported results. In particular, the improper application of data augmentation prior to dataset splitting — a practice observed in precisely one-third of the reviewed studies — poses a significant risk of data leakage and can result in overly optimistic performance estimates. Additional challenges include inconsistent evaluation metrics, limited dataset diversity, and a lack of standardized benchmarks.
This review provides a structured overview of current approaches, a critical assessment of existing research practices, and actionable guidance for developing robust, deployment-ready DL solutions for precision agriculture.

Keywords: Crop Harvesting, Strawberry, Machine Vision, Object Detection, Fruit Detection, Ripeness Classification, Picking Point Localization, Deep Learning Models, YOLO

1 Introduction

Strawberries are known for their high nutritional value, i.e., they are rich in vitamins C, B1, B2, and carotene, and have significant economic potential in agricultural production. They are considered healthy and contain relatively little sugar despite their sweet taste. The global strawberry industry was valued at over $22 billion in 2024, and its economic impact extends well beyond the farm: In addition to fresh sales, the industry generates substantial revenue through products such as jams, frozen desserts, smoothies, and cosmetics [1].

Since strawberries do not ripen after being picked, harvesting them requires accurate detection and precise determination of their ripeness. Moreover, harvesting strawberries is one of the most labor-intensive tasks in agriculture. Each fruit must be hand-picked when it is perfectly ripe, handled carefully, and processed promptly. This requires a significant investment of personnel, time, and resources from farms [2]. As the global population continues to age, the labor shortage in agriculture is becoming increasingly acute, leading to rising production costs, particularly in industrialized countries. In this context, the development and implementation of automated fruit harvesting technologies are strategically vital for enhancing production efficiency, enabling round-the-clock operations, and addressing labor shortages [3, 4].

In recent years, agricultural technologies have received significant attention in scientific research, with many efforts to develop them, particularly regarding the application of robots and automation in intensive crop production [5]. Using robotics in protected agriculture would increase overall performance and production efficiency while improving labor quality and safety. The ability of autonomous harvesting robots to pick the right fruit at the right time depends directly on the accuracy of ripeness detection [6].

Deep learning (DL), a branch of artificial intelligence, is increasingly recognized as a transformative tool in this field. DL models can automate complex tasks such as crop monitoring, yield estimation, disease diagnosis, and pest management with high accuracy by leveraging large datasets of images, sensor readings, and other farm data. Despite the promise of DL in agriculture, gaps remain in understanding its full capabilities and limitations in real-world farming. [7]

While two-stage detectors like Faster R-CNN and Mask R-CNN were historically dominant in agricultural applications due to their high detection accuracy [8, 9], their computational complexity and limited inference speed pose challenges for real-time robotic harvesting [3]. In contrast, one-stage detectors, particularly the YOLO (You Only Look Once) family, have gained significant traction by framing object detection as a single regression problem, enabling substantially faster inference while maintaining competitive accuracy. This architectural shift is reflected in the literature: Charisis and Argyropoulos [8] found that out of 77 DL-based instance segmentation papers in agriculture (2019–2023), 56 employed two-stage architectures like Mask R-CNN, while only 11 used one-stage algorithms such as YOLO. However, a more recent bibliometric analysis by Pugazhendi et al. [10] of 620 articles on YOLO-based fruit detection (2020–early 2025) demonstrates the rapid ascendancy of one-stage methods. Similarly, Ma et al. [11] reviewed 24 DL-based fruit ripeness detection papers and found that 20 utilized one-stage detectors (primarily YOLOv3–v8 and YOLOX), compared to only 4 using two-stage CNNs. This trend underscores a broader transition in agricultural computer vision: from region-proposal-based pipelines toward streamlined, real-time-capable architectures that are better suited for deployment on autonomous harvesting platforms.

Several surveys and reviews [12–19] have addressed certain aspects of this rapidly evolving field. However, most of these reviews offered a more expansive perspective on the evolution and application of DL for object detection, such as in [12, 14, 18], or primarily focused on detecting plant diseases or pests, such as in [13, 16, 17, 19].

Only a few available review papers [8, 11, 20, 21] address, at least in part, crop detection for agricultural monitoring or harvesting. The survey by Muthulakshmi and Renjith [20] is a relatively brief review of just 15 papers, published between 2007 and 2021, on the classification of multiple fruits (banana, cherry, mango, durian, etc.) with both machine learning (ML) and DL techniques. Charisis and Argyropoulos [8] provided an overview of 77 DL-based instance segmentation implementations in agriculture from 2019 to 2023. They considered a wide range of crop-related agricultural tasks, including crop phenotyping, health status monitoring, yield prediction, disease or weed detection, and robotic applications. 62% of the papers included in the survey addressed crop detection, considering species such as apples, tomatoes, lettuce, strawberries, and grapes. They stated that the majority of the papers (56) implemented two-stage CNN detector architectures, such as Mask R-CNN. Only 11 papers were found that used one-stage detection algorithms like YOLO. Ma et al. [11] presented a systematic survey of advances in fruit ripeness detection and classification using spectral analysis (12 papers from 2016 to 2024), traditional image processing (9 papers from 2014 to 2023), and DL techniques (24 papers from 2015 to 2024). They focused on a wide variety of crop species, including apples, cherries, strawberries, and mangoes. The review included 4 papers on two-stage CNNs for fruit maturity detection, with a primary focus on Faster R-CNN and Mask R-CNN. On the other hand, 20 papers using one-stage detectors, primarily YOLOv3–v8 and YOLOX, were analyzed. Our recent review in [21] provided a systematic review of object detection in plants (e.g., crops, diseases, and growth stages) but focused on analyzing architectural modifications introduced in improved YOLOv8 algorithms.

The very recent review by Pugazhendi et al. [10] provides a comprehensive bibliometric analysis of literature on YOLO-based agricultural fruit detection. The researchers examined 620 articles from January 2020 to early 2025 and extracted 148 relevant records. The work showcased a variety of fruits, spanning over 30 species, and covered three stages of application: pre-harvest monitoring, harvesting operations, and post-harvest processing. Their analysis of YOLO versions and the architectural modifications employed for fruit detection in different application scenarios, with regard to various fruit characteristics and environmental conditions, is of particular value. Badgujar et al. [15] conducted an earlier work involving a systematic, bibliometric review of 257 publications on YOLO applications across general agricultural domains, from 2018 to November 2023. The authors also identified knowledge gaps, critical gaps, and modifications to YOLO (v3–v7) for specific agricultural tasks through analysis of 30 selected articles. Xiao et al. [9] presented a review of 56 papers that used DL for the detection and recognition of fruits such as apples, tomatoes, and citrus in the context of automatic harvesting, from 2017 to 2022. They found that during the period under consideration, research was dominated by two-stage fruit detection and recognition methods, such as Faster R-CNN and Mask R-CNN.

While several reviews [8–13, 15–21] have addressed certain aspects of this rapidly evolving field, most offer a broader perspective on agricultural DL or focus on different tasks such as disease or pest detection. For instance, the systematic review by Chettri et al. [7] provides a comprehensive methodological analysis of precision agriculture in general but does not specifically address the unique challenges of strawberry fruit and ripeness detection.

To the best of our knowledge, none of these reviews provides a comprehensive analysis specifically focused on this task, which is the main scope of the present work. This dedicated focus on the strawberry is motivated by a unique constellation of challenges that set it apart from other crop detection tasks, positioning it as an ideal yet exceptionally demanding 'model organism' for fine-grained agricultural object detection. Unlike many fruits with distinct color transitions, strawberries exhibit red pigmentation across a wide developmental window, turning ripeness labeling into a highly ambiguous problem. Their growth in dense, ground-level clusters creates occlusion scenarios that challenge detectors far beyond what is typical for tree fruits. And as non-climacteric and fragile fruits, they demand precise, damage-free picking-point identification, linking detection accuracy directly to robotic harvesting success. Together, these challenges — label ambiguity, extreme occlusion, and precision beyond detection — form a uniquely demanding benchmark that serves as a stress test for agricultural computer vision.

The contributions of this review can be summarized as follows:

- We provide the first structured and focused overview of how DL-based models, represented by the YOLO family (YOLOv5–v11), and hybrid models, i.e., CNNs + ViTs, have been applied to detecting strawberry fruit and assessing ripeness. We highlight the performance these models have achieved and the specific challenges they address, thereby demonstrating DL's potential to automate elementary steps in agricultural monitoring and harvesting tasks.
- We conduct a comparative analysis of methodological choices across 60 reviewed studies published between 2023 and May 2026. This includes comparing defined ripeness classes, dataset characteristics, preprocessing and augmentation strategies, and evaluation protocols. By analyzing these aspects, we shed light on current trends and best practices. On the one hand, we propose a structured analysis framework to systematically assess the architectural modifications and their impact on model performance. On the other hand, we identify areas that require more attention, such as the sequence in which datasets are split and augmented to avoid data leakage and overly optimistic performance evaluations.
- We discuss the challenges and open issues hindering the deployment of DL models in real-world farming environments and connect them to future research directions. Key challenges include data availability and diversity, a lack of standardized benchmarks, the high risk of data leakage due to methodological flaws such as augment-then-split protocols, the inductive bias, and the data hunger of Transformer models. Through these contributions, our review synthesizes the current state of strawberry fruit detection, ripeness classification, and picking point localization using DL, providing actionable insights to guide researchers and practitioners in advancing the field.

2 Scope and methodology of the review

2.1 Keyword identification and database search

This review of DL-based strawberry detection and ripeness classification studies was largely based on the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) guidelines [22]. First, a comprehensive literature search was conducted in Scopus, IEEE Xplore, and Google Scholar to identify relevant peer-reviewed articles and conference papers. The following search terms "*detection/recognition*", "*picking point/s*", "*pose*", "*key point/s*", "*mature/ripeness*", and "*strawberry/strawberries*" in combination with Boolean operators AND and OR. Second, each identified study underwent a detailed full-text review and critical evaluation against predefined inclusion and exclusion criteria. From these queries, more than 200 studies were retrieved from the databases. The flow chart of the subsequent (PRISMA) selection process is shown in Figure 1.

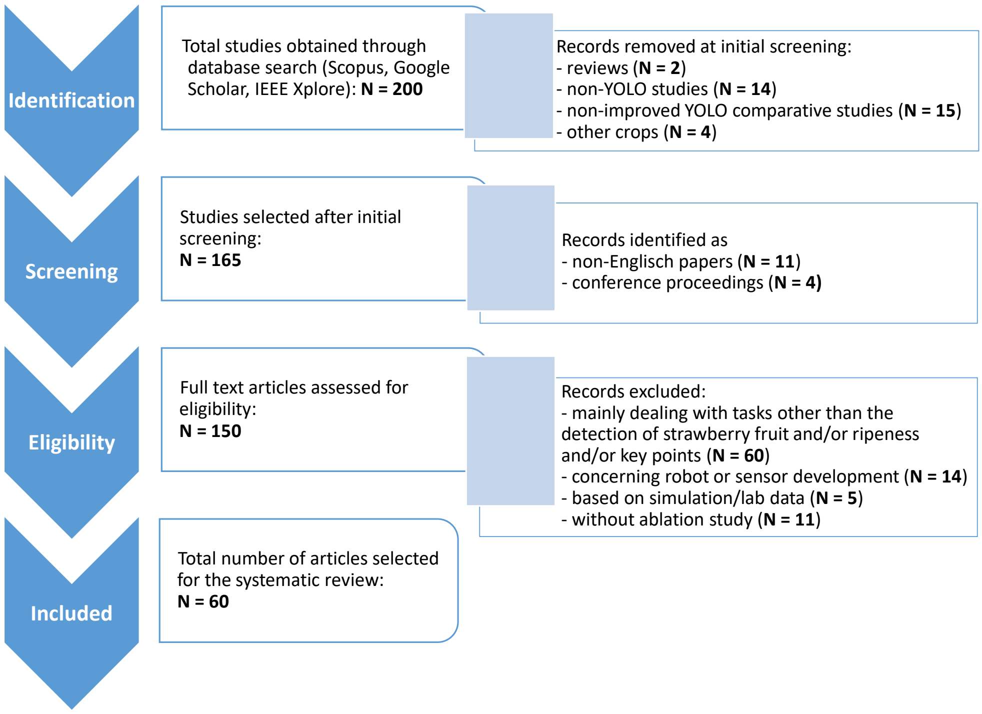

Fig. 1: Flow chart illustrating the selection process of the 60 studies included in the systematic review

2.2 Inclusion criteria

In order to find the relevant studies to the scope of this review, each paper selected was analyzed in detail, taking into account the following inclusion criteria:

- Paper was published between 2023 and May 2026 and describes an improved YOLO-based DL object detection algorithm.
- Study concerns a crop-related harvesting task, i.e., detection of fruit or ripeness, including segmentation or key point detection.
- A strawberry-related study is presented.
- A dataset is collected using machine vision, self-built by combining different data sources, or a public dataset is used.
- Quantitative performance evaluation metrics (P, R, mAP, FPS, Params, GFLOPs, etc.) or qualitative explainable AI methods are used, and their results are presented.
- Paper includes an ablation study or an equivalent explanation of how the model improves upon the baseline or other ML/DL algorithms.
- Paper is written in the English language.

2.3 Exclusion criteria

The exclusion criteria are as follows:

- Review papers and studies that primarily aim to compare different versions of the original YOLO are excluded.
- Disease detection and yield prediction studies, as well as other studies dealing with tasks other than the detection of strawberry fruit and their ripeness, are excluded.
- Papers that mainly concern the development of robots, sensors, or similar aggregates are excluded.
- Studies on other crops than strawberries are excluded.
- Papers using only simulation data are excluded.
- Papers that are not accessible via institutional licenses, not peer-reviewed, or unavailable online, are excluded.
- Papers in languages other than English are excluded.

2.4 Selection process

An initial screening of the search results was conducted based on the titles and abstracts of each study. Studies that appeared irrelevant or did not meet the inclusion criteria were excluded. This preliminary screening reduced the total number of studies to 165 for detailed evaluation.

The remaining studies underwent a full-text review to assess their relevance, methodological rigor, and contribution to the field. During this stage, we applied the inclusion and exclusion criteria in detail, evaluating each study's objectives, methodology, findings, and overall quality. After full-text reviews, 60 articles were selected for the final review and are included in this work. Table 2 provides a comparative overview of the DL architectures, datasets, and performance evaluation results of the selected studies [23–82]. These results are presented in detail in Sections 3 and 4.

Table 1: Summary of the publicly available strawberry datasets and their frequency of usage across the reviewed studies

Dataset	Size (N)	Details	Frequency
"Self-built"	Varying N, from 247 to 7,109	**Characteristics:** Images collected from an agricultural environment or taken from the Internet; **Annotations:** individual; **Challenges:** Proprietary, often small dataset, data imbalance	53
StrawDI / StrawDI_Db1 [83]	3,100	**Characteristics:** Collected from plantations, Spain; **Annotations:** 17,938 Bounding Boxes (BBs) of individually labeled strawberries; **Challenges:** Strawberries that are imperceptibly small, partially occluded by leaves or other fruit, positioned at image edges, or misshapen	8
Strawberry-DS [84]	247	**Characteristics:** Collected from orchard at lab, Egypt; **Annotations:** 1,062 BBs; **Challenges:** Overlapping of fruits with different shapes and sizes	1
Largest Strawberry Dataset (LSDS)	$N = N1 + N2 + N3$	**Characteristics:** Combining N1 images from StrawDI_Db1 + N2 images from public platforms (Internet) such as Kaggle or Roboflow + N3 self-captured strawberry images	3
[39]: $5{,}600 = 2{,}700 + 1{,}500 + 1{,}400$		**Annotations:** 75,505 labels	
[40]: $5{,}600 = 2{,}700 + 1{,}500 + 1{,}400$		**Annotations:** $>90{,}000$ labels	
[69]: $5{,}500 = 2{,}500 + 1{,}200 + 1{,}800$		**Annotations:** 66,913 labels	
[85]: $7{,}700 = 3{,}100 + 2{,}500 + 2{,}100$		**Annotations:** 85,601 labels	
[86]: $5{,}600 = 2{,}800 + 1{,}600 + 1{,}200$		**Annotations:** 31,174 labels; **Challenges:** Morphological diversity of strawberries, small and dense targets, and high overlap scenes	
Final Strawberry Ripeness [87]	516	**Characteristics:** Good quality images of strawberries that were categorized into pickable and unpickable classes; **Annotations:** 516 BBs (either pickable or unpickable fruit); **Challenges:** Changing lighting conditions, occlusions by leaves, dirt, and perspective distortions	1
New Strawberries [88, 89]	3,100	**Characteristics:** Collected from a greenhouse, China; **Annotations:** 16,123 BBs; **Challenges:** Fine-grained grades of maturity and different growth stages	1

3 Background: data management and preparation

The accuracy and performance of DL models depend significantly on the quality, diversity, and distribution of the data within the dataset. Insufficient variability can lead to bias, overfitting, and limited generalizability. The quality of the dataset is particularly important for strawberry detection and ripeness assessment, as environmental factors such as varying lighting conditions and occlusion can complicate detection. This chapter describes the approaches used in the reviewed studies for data collection, annotation, preprocessing, and dataset distribution. In addition, strategies for detecting strawberry fruit and ripeness, as well as risks such as data leakage, are discussed.

3.1 Strawberry detection and ripeness stages

In the domain of smart farming, crop detection uses machine vision and artificial intelligence to automatically identify, locate, and analyze agricultural targets in image or video data. Digital images are important data sources that provide precise, region-specific information to address a variety of agricultural problems [8]. To capture this information, computer vision systems employ a variety of sensing modalities depending on the specific application. For instance, RGB imaging is employed for visual inspection, multispectral or hyperspectral imaging for the detection of biological processes, and ranging sensors for geometric measurements [90]. The detection task is typically achieved by generating BBs that delineate the regions where objects are detected [17]. Crop detection must cope with highly unstructured agricultural environments where targets vary significantly in size, shape, and color, and are often partially occluded by leaves, stems, or other fruits [33]. The primary objective is to provide precise spatial information in the form of BBs or segmentation masks. This information enables a variety of precision agriculture applications, including the detection of fruit and vegetables for harvesting operations such as yield estimation and robotic harvesting, as well as the detection of crop diseases for early intervention and growth stage monitoring to optimize field management [91, 92].

The process of crop detection typically follows a structured pipeline comprising several stages: preprocessing, feature extraction, learning, and classification [93]. The power of DL-based techniques lies in their ability to automatically absorb the most relevant attributes from input data, making them particularly well-suited for complex agricultural environments where manual feature extraction is challenging [93]. This capability has led to the widespread adoption of convolutional neural networks (CNNs), which have evolved to become the most powerful approach for solving target identification and classification problems in agriculture [94].

For strawberry harvesting in particular, accurate fruit detection is critical. Fruit instance segmentation is a key element for autonomous picking systems, and the precise localization of each strawberry directly affects the viability of the entire harvesting operation [95]. This need for precision is further emphasized by He et al. [23], who highlight that detection accuracy in strawberry farming is severely affected by overlapping foliage, variable lighting conditions, and fruits at different growth stages. Moreover, crop detection for harvesting applications requires more than just identifying the presence of a fruit. It demands precise localization to enable subsequent operations such as

Table 2: A comparative table of the architectures, data preprocessing steps, and the evaluation results reported in the identified studies for the detection and classification of strawberry fruit or ripeness.
(Symbol legend: ● = The data augmentation technique was applied appropriately. ● = The data augmentation technique was apparently applied before data splitting, implying *Potential Data Leakage* and overoptimistic performance metrics. ○ = No augmentation reported, – = N/A., Split ratio = training:validation:testing set ratio.)

Ref.	**Architecture**	**Dataset Characteristics**	**Preprocessing/ Augmentation**	**Evaluation Metric(s)**	**mAP Gain/Loss**
			YOLOv5		
[23]	Improved lightweight YOLOv5	StrawDI_DB1 3,100 images	○ split ratio: 8:1:1	mAP: 97.23% Params: 1.45M FPS: 17.74 GFLOPs: 3.4	+1.10%
[24]	MS-YOLOv5	Self-built 3,468 images	● 9,536 images split ratio: 7:1:2	mAP: 95.6% Params: 7.02M FPS: 76.0 GFLOPs: 12.9	**+8.40%**
[25]	YOLOv5-ASFF	Self-built 1,217 images	○ 9,225 instances split ratio: 7:1:2	mAP: 91.86% FPS: 56	+0.99%
[26]	YOLOv5s-BiCE	Self-built 1,446 images	● 2,892 images 16,000 instances split ratio: 7:2:1	mAP: 96.10% FPS: 51	+2.70%
[27]	YOLOv5s-CGhostNet	Self-built 2,000 images	● online 10,450 instances split ratio: 8:1:1	mAP: 91.70% Params: 6.02M FPS: 64.0 GFLOPs: 9.8	+1.00%
[28]	RTF-YOLO (Improved YOLOv5s)	Self-built 2,040 images	● 9,180 images 8,712 instances split ratio: 7:2:1	mAP: 89.21% Params: 13M FPS: 145.0	+3.69%
[29]	CCEK-YOLOv5s	Self-built 1,047 images	● 11,517 images (overlap check) split ratio: 8:1:1	mAP: 97.90% Params: 9.47M FPS: 64.93 GFLOPs: 22.4	+0.40%
[30]	YOLOv5s-Straw	Self-built 1,540 images	● 10,631 instances split ratio: 9:1	mAP: 80.70% Params: 9.40M FPS: 55.25 GFLOPs: 20.4	+2.90%
			YOLOv7		
[31]	DSW-YOLO (Improved YOLOv7)	Self-built 1,538 images	○ split ratio: 8:2:0 (no test data)	mAP: 86.70% Params: 32.40M FPS: 42.0 GFLOPs: 99.5	+2.20%
[32]	Improved YOLOv7	Strawberry-DS 247 images	● 1,235 images split ratio: 6:2:2	mAP: 83.10% Params: 8.46M GFLOPs: 24.1	+3.50%
[33]	Improved YOLOv7t	Self-built 1,145 images	● 3,600 images split ratio: 7:1:2	mAP: 89.80% Params: 11.7 FPS: 357.1 GFLOPs: 5.8	+2.60%

Table 2 (continued)

Ref.	Architecture	Dataset Characteristics	Preprocessing/ Augmentation	Evaluation Metric(s)	mAP Gain/Loss
[34]	Improved YOLOv7	Self-built 4,393 images	• online split ratio: 8:1:1	mAP: 92.10% Params: 34.53M GFLOPs: 109.9	+4.60%
[35]	MFD-YOLO (Improved YOLOv7t)	Self-built 2,370 images	• 12,970 images split ratio: 7:1:2	mAP: 97.50% Params: 1.74M FPS: 125.1 GFLOPs: 3.0	0.00%
			YOLOv8		
[36]	HSK-YOLO (Improved YOLOv8)	Strawberry Recognition 607 images	• 1,214 images split ratio: 7:2:1	mAP: 93.00% Params: 2.21M GFLOPs: 7.2	**+5.40%**
[37]	YOLOv8+	Self-built 1,187 images	○ split ratio: 8:1:1	mAP: 91.91% FPS: 83.33	+1.27%
[38]	CES-YOLOv8	Self-built 546 images	• 2,722 images split ratio: 4:0:1	mAP: 92.10% FPS: 184.52	+2.00%
[39]	HM-YOLO (Improved YOLOv8)	LSDS: N = 5,600 (N1 = 2,700; N2 = 1,500; N3 = 1,400) images	• 16,800 images 77,505 instances split ratio: 6:2:2	mAP: 92.70% Params: 4.10M FPS: 87.5 GFLOPs: 11.4	+1.70%
[40]	RCEA-YOLOv8	LSDS: N = 5,600 (N1 = 2,700; N2 = 1,500; N3 = 1,400) images	• 16,800 images 90,000 instances split ratio: 6:2:2	mAP: 94.90% Params: 3.90M	+2.30%/ +1.00%
[41]	YOLOv8n-RML	Self-built, – images	• 9,305 images 22,985 instances split ratio: 8:1:1	mAP: 91.30% Params: 1.50M GFLOPs: 5.3	–0.30%
[42]	Improved YOLOv8	Self-built 1,680 images	• 3,500 images split ratio: 8:1:1	mAP: 93.10% Params: 57.48M	**+13.47%**
[43]	YOLOv8s-FEGW	Self-built 2,000 images	• 7,250 images split ratio: 8:1:1	mAP: 93.80% Params: 7.20M FPS: 212.0 GFLOPs: 19.2	+2.70%
[44]	YOLOv8-CDW	Self-built 1,000 images	• 3,713 images split ratio: 8:1:1	mAP: 97.3%	+1.10%
[45]	Improved YOLOv8	Self-built 1,021 images	• 5,105 images 14,910 instances split ratio: 6:2:2	mAP: 96.40% Params: 1.30M GFLOPs: 4.4	**+5.00%**
[46]	Improved YOLOv8s	Self-built 2,190 images	• 4,250 images 13,228 instances split ratio: 8:1:1	mAP: 92.50% Params: 8.00M FPS: 103.0 GFLOPs: 27.1	+0.40%
[47]	GAM-YOLOv8s	Self-built 1,500 images	• 6,000 images split ratio: 8:1:1	mAP: 91.50% FPS: 53.0	+4.60%

Table 2 (continued)

Ref.	Architecture	Dataset Characteristics	Preprocessing/ Augmentation	Evaluation Metric(s)	mAP Gain/Loss
[48]	YOLOv8-SP	Self-built 2,800 images	● 5,600 images split ratio: 7:2:1	mAP: 95.40% Params: 1.86M GFLOPs: 5.7	+3.40%
[49]	STRAW-YOLO (Improved YOLOv8)	Self-built 3,262 images	○ split ratio: 8:2:0 (no test data)	mAP: 96.00% Params: 15.30M FPS: 62.6	+2.50%
[50]	RLK-YOLOv8	Self-built 3,628 images	● online 11,182 instances split ratio: 8:1:1	mAP: 95.40% FPS: 33	+2.10%
[51]	WCS-YOLOv8s	Self-built 1,957 images	● online split ratio: 7:2:1	mAP: 87.53% Params: 18.69M FPS: 45.9	+1.94%
[52]	Improved YOLOv8s-pose	Self-built 2,252 images	● 3,000 images split ratio: 7:1:2	mAP: 97.10% Params: 4.74M GFLOPs: 14.4	+0.60%
[53]	YOLOv8s-seg-CBAM	Self-built 4,050 images	○ split ratio: 8:1:1	mAP: 86.2%	+2.50%
[54]	Improved YOLOv8	Self-built 959 images	● 3,264 images split ratio: 5:1:1	mAP: 92.10% FPS: 58.1	+1.60%
[55]	DS-YOLO (Improved YOLOv8)	Self-built 800 images	● 1,920 images 7,924 instances split ratio: 8:1:1	mAP: 97.10% Params: 1.81M GFLOPs: 4.9	+1.60%
[56]	YOLOv8n-MCP	StrawDI_Db1 3,100 images	● online augmentation split ratio: 8:1:1	mAP: 75.00% Params: 1.00M FPS: 96.15 GFLOPs: 5.3	–2.00%
[57]	Improved YOLOv8s + YOLOv5s-cls	Self-built 1,600 images	● split ratio: 7:1	mAP: 83.20% FPS: 119.1	+2.50%
[58]	Improved YOLOv8n + IMU	Self-built 9,600 images	● split ratio: 6:2:2	mAP: 98.56% Params: 2.18M FPS: 30 GFLOPs: 6.2	+0.19%
YOLOv9					
[59]	ASHM-YOLOv9	Self-built 670 images	● 2,682 images 13,668 instances split ratio: 8:1:1	mAP: 99.10% GFLOPs: 253.3	+0.70%
[60]	YOLOv9-CS	Self-built 800 images	○ split ratio: 8:1:1	mAP: 96.61% FPS: 64.0	+4.16%
[61]	CR-YOLOv9	Self-built 4,970 images	○ split ratio: 7:2:1	mAP: 97.95% FPS: 84.0	+3.34%
[62]	SGSNet (Improved YOLOv9n)	Self-built 660 images	● 7,528 images split ratio: 8:1:1	mAP: 99.50% Params: 5.86M GFLOPs: 14.7	+0.25%

Table 2 (continued)

Ref.	Architecture	Dataset Characteristics	Preprocessing/ Augmentation	Evaluation Metric(s)	mAP Gain/Loss
			YOLOv11		
[63]	YOLOv11-GSF	Self-built 2,985 images	• split: 6:3:1 ratio	mAP: 97.30% FPS: 67.5 GFLOPs: 6.0	+0.50%
[64]	LBS-YOLO (Improved YOLOv11n)	New Strawberries 3,100 images	• 4,000 images split ratio: 7:2:1	mAP: 88.60% Params: 1.6M FPS: 260.7 GFLOPs: 6.6	+2.20%
[65]	YOLOv11-HRS	Self-built 7,109 images	○ 65,901 instances split ratio: 8:1:1	mAP: 89.80% Params: 2.13M FPS: 252.5 GFLOPs: 9.7	+3.40%
[66]	YOLOv11-LES	Self-built 801 images	• 2,724 images 3,586 instances split ratio: 8:1:1	mAP: 86.50% Params: 2.14M FPS: 65.1 GFLOPs: 6.1	+3.20%
[67]	YOLO-RCMC	Self-built 2,223 images	• split ratio: 7:2:1	mAP: 89.00% Params: 17.6M FPS: 109.9 GFLOPs: 54.7	+3.80%
[68]	MVYOLO-Tiny (YOLOv11 + MobileViTv2)	Self-built + StrawDI 3,959 images	• 14,248 instances split ratio: 2.4:6.6:1.0	mAP: 94.00% Params: 1.19M FPS: 94 GFLOPs: 5.3	+0.60%
[69]	SR-YOLO (Improved YOLOv11)	LSDS: N = 4,500 (N1 = 2,500; N2 = 1,200; N3 = 1,800) images	• 14,300 images 66,913 instances split ratio: 6:2:2	mAP: 92.10% Params: 2.3M FPS: 90.1 GFLOPs: 5.8	+2.50%
[70]	DHN-YOLO (Improved YOLOv11)	Self-built 2,066 images	• 5,018 images split ratio: 8:1:1	mAP: 89.70/91.80% Params: 1.90M FPS: 71.6	+0.90%
[71]	RAFS-YOLO (Improved YOLOv11)	Self-built 926 images	• online 8,627 instances split ratio: 8:1:1	mAP: 94.60% Params: 1.63M FPS: 4.66 GFLOPs: 4.8	+2.00%
[72]	B2G-YOLOv11-S	Self-built, 5,418 images	○ 17,342 instances split ratio: 8:1:1	mAP: 95.60% FPS: 52.36 GFLOPs: 13.8	**+6.30%**
[73]	Improved YOLOv11-seg	Self-built 1,868 images	• 4,500 images split ratio: 8:1:1	mAP: 92.90%	+1.29%
[74]	YOLOv11-SCC	Self-built, 5,343 images	○ 12,000 instances split ratio: 8:1:1	mAP: 94.30% Params: 2.62M FPS: 42.2 GFLOPs: 6.7	+2.70%
[75]	SML-YOLO (Improved YOLOv11s)	Self-built, 1,075 images	• 3,305 images 14,454 instances split ratio: 7:2:1	mAP: 92.40% Params: 6.49M FPS: 256.41 GFLOPs: 15.0	+2.90%

Table 2 (continued)

Ref.	Architecture	Dataset Characteristics	Preprocessing/ Augmentation	Evaluation Metric(s)	mAP Gain/Loss
			CNNs+ViTs		
[76]	YOLOv5s + ViT	Self-built 440 images	• split ratio: 8:0.4:1.6	mAP: 92.90% Params: 7.02M	**+9.30%**
[77]	LS-YOLOv8s (LW-Swin)	Self-built 1,089 images	• 7,515 images split ratio: 8:1:1	mAP: 94.40% mAP: 98.70% (with augm.) FPS: 19.23 GFLOPs: 48.8	**+5.20%**
[78]	SDSLP-YOLO (YOLOv8-pose + SwinT)	StrawDI_DB1 3,100 images	• 17,980 instances split ratio: 7:2:1	mAP: 88.30% Params: 3.66M	+1.70%
[79]	Improved YOLOv8-pose (MobileViTv3)	Self-built 3,860 images	○ split ratio: 7:2:1	mAP: 97.85% Params: 3.65M FPS: 58.48 GFLOPS: 4.59	**+5.49%**
[80]	DynaFog-BerryNet (Improved YOLOv8)	StrawDI 8,000 images	○ split ratio: –	mAP: 95.07%	+0.50%
[81]	DPViT-YOLOv8	Self-built 1,198 images	• split ratio: 9.98:1:1	mAP: 77.90% Params: 4.10M FPS: 94.5 GFLOPs: 10.3	+1.70%
[82]	Improved YOLOv9 (SwinT)	Self-built 2,000 images	○ split ratio: 8:1:1	mAP: 87.30%	+1.20%

picking-point determination, particularly when fruits are clustered or overlapping [85]. In addition to the fruit detection and classification of fruit ripeness, autonomous harvesting systems additionally require the precise localization of a picking point, which is typically on the stem or calyx, to enable damage-free robotic grasping.

Object detection is formally defined as a two-step task: (i) object localization, which identifies the presence and spatial extent of objects of interest by generating BBs, and (ii) object classification, which assigns each detected object to a predefined class (e.g., ripeness stage) [96].

In summary, crop detection for strawberries encompasses several key elements: localization, which involves determining the position and extent of each fruit through BBs or segmentation masks [17, 23, 95]; robustness, which requires maintaining detection accuracy despite challenges such as occlusion, scale variation, and changing illumination [8, 33, 94]; multi-stage processing, following the established pipeline of preprocessing, feature extraction, learning, and classification [17, 93]; and downstream integration, providing outputs that facilitate subsequent tasks like ripeness classification, counting, quality assessment, or robotic harvesting [48, 85, 94].

A detailed review of the DL architectures employed for these tasks, including YOLO-based detectors and hybrid models, is presented in Section 4.

Strawberries go through several stages of development. The process from flower to fruit can be divided into the flowering stage, the fruit development stage, the ripening process, and the harvesting stage. The color of the strawberry depends on the variety, especially during ripening. Some strawberries may take on yellow or pink hues. Since strawberries do not mature after harvesting, the recommended harvest time is, according to ISO 6665:1983 (1983), defined as the point at which a strawberry exhibits at least 75% of the variety-specific red color. This definition of the harvest time is based on a balance between shelf life and optimal ripeness for consumption. Although there is a recommended harvest time, there is no standardized definition of ripeness [64].

To determine the level of maturity using image-based methods, two approaches are typically used:

- Classification of the degree of ripeness into individual categories, such as unripe, semi-ripe, ripe, and overripe.
- Quantification by measuring the proportion of red pigments of the strawberry.

Table 3 summarizes the classification schemes adopted across the reviewed studies, revealing a striking heterogeneity: the number of ripeness classes ranges from binary classification (ripe vs. unripe) in [37, 97] to six-category systems as in [32]. Figure 2 illustrates representative examples.

This heterogeneity reflects a deeper, fundamental methodological challenge that distinguishes strawberry ripeness detection from generic object detection: strawberry ripening is a phenological continuum (green $\rightarrow$ white $\rightarrow$ turning $\rightarrow$ red $\rightarrow$ overripe), not a set of discrete, objectively separable states. Despite the existence of ISO 6665:1983 for harvest timing, there is no universally accepted, standardized definition of ripeness categories [64]. The reviewed studies define class boundaries on an ad-hoc basis, relying on red surface ratio thresholds (e.g., 0–30%, 30–80%, 80–100% [24, 27]) or qualitative descriptions (e.g., 'turning,' 'mature' [64]).

This continuous-to-discrete mapping has direct consequences for model evaluation and real-world applicability. First, the practical definition of class boundaries is inherently subjective and observer-dependent. Distinguishing a 'partially ripe' strawberry from a 'ripe' one, for example, is a judgment call rather than an objective measurement. Across the 60 reviewed studies, none reported quantitative inter-annotator agreement measures (e.g., Cohen's κ) or systematic labeling consistency protocols. The absence of such validation means that inconsistent ground truth labels can directly propagate into unreliable performance metrics. This problem is most acute for fruit at transitional stages of ripeness, where visual ambiguity is greatest. Consequently, per-class performance metrics (e.g., AP per ripeness category) are essential for disambiguating whether observed mAP improvements stem from better detection of clear-cut cases or from genuine progress in resolving these boundary regions. However, such per-class breakdowns are reported in only a minority of the reviewed studies, making it difficult to assess true progress on the most challenging transitional cases.

3.2 Performance metrics for object detection

Object detection models are commonly evaluated using three categories of metrics: *accuracy metrics* that measure detection and localization quality, *speed metrics* that assess real-time suitability, and *efficiency metrics* that quantify computational resource demands [98].

3.2.1 Accuracy metrics

In the object detection literature, precision (P), recall (R), and F1-score (F1) are among the most widely reported metrics, as they provide a balanced assessment of detection accuracy. These metrics are derived from the confusion matrix, where true positives (TP) are correct detections, false positives (FP) are incorrect detections, and false negatives (FN) are missed objects [99]:

$$\text{Precision} = \frac{TP}{TP+FP}, \quad \text{Recall} = \frac{TP}{TP+FN}, \quad \text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1}$$

Mean Average Precision (mAP) has emerged as the primary benchmark metric for comparing model performance, as it integrates precision and recall across varying confidence thresholds [8, 100]. Average Precision (AP) is calculated as the area under the precision-recall curve, and mAP is the mean of AP across all object classes [101].

A critical parameter is the *Intersection over Union* (IoU) threshold, which defines when a predicted bounding box matches a ground truth annotation:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{2}$$

Two common variants exist:

- mAP@0.5: Uses a fixed IoU threshold of 0.5. A prediction is considered correct if its overlap with the ground truth exceeds 50% [37, 60, 82].
- mAP@0.5:0.95: Averages AP over multiple IoU thresholds from 0.5 to 0.95 in steps of 0.05, penalizing poor localization more severely [101, 102].

Comparing model performance across studies remains challenging due to inconsistent use of these thresholds. In strawberry detection, mAP@0.5 is the most commonly reported metric, aligning with standard practice in agricultural object detection tasks. Nevertheless, using mAP@0.5:0.95 is recommended for more rigorous evaluation.

3.2.2 Speed metrics

For real-time applications such as robotic harvesting, inference speed is critical. The most common metric is *Frames Per Second* (FPS), which indicates how many images a model can process per second. However, FPS is highly dependent on the underlying hardware (GPU model, precision, batch size). Some studies instead report *inference time* in milliseconds (ms) per image, which is more informative when the batch size is set to 1 [37].

In the literature, the following distinctions are commonly made, though not always consistently applied [102]:

- **Inference time**: Core model execution time only.
- **Latency**: End-to-end processing time including preprocessing and postprocessing.
- **FPS**: Throughput metric derived from inference time (see Eq. 3).

The conversion between inference time t_{inf} (in ms) and FPS is given by:

$$\text{FPS} = \frac{1000}{t_{\text{inf}}} \tag{3}$$

In practice, some studies report latency synonymously with inference time or calculate FPS from end-to-end latency. The definitions above follow the stricter convention recommended for reproducible benchmarking.

3.2.3 Efficiency metrics

To measure model efficiency independently of hardware, two metrics are commonly used:

- **Parameter count (Params)**: The number of trainable weights, correlating with memory footprint [103]
- **GFLOPs** (Giga Floating-Point Operations): The number of floating-point operations required for a single forward pass, providing a hardware-agnostic estimate of computational complexity [102]

Lower GFLOPs generally indicate better suitability for edge devices such as the NVIDIA Jetson series [104].

3.3 Data collection

The first step in detecting and classifying strawberry fruit ripeness is data collection. A number of datasets, including the STRAW_DI/StrawDI_Db1, Strawberry-DS, Final Strawberry Ripeness, LightStrawberry, Open-Field Strawberry, and Strawberry Recognition Dataset, have been used in the reviewed studies. Table 1 gives a quick overview of these datasets. Most of the datasets used are self-constructed from agricultural fields. Some researchers [39, 40, 69] have built so-called Largest Strawberry Datasets (LSDS) by combining images from StrawDI_Db1, images from public platforms (Internet) such as Kaggle and Roboflow, and self-captured strawberry images. This combined approach overcomes the limitations of other datasets, which are typically small in scale, lack scene diversity, and have imprecise annotations [85]. Large datasets improve the model's ability to generalize across different cultivation conditions. However, to maintain accuracy and data integrity in object detection tasks, duplicate images must be avoided [69, 105].

Table 2 (Columns 3 and 4) provides a comparative overview of dataset characteristics and whether data augmentation was applied in the studies examined, with a focus on potential data leaks.

Table 3: Overview of strawberry ripeness classification schemes grouped by number of classes

Classes	Reference	Description
1	[81]	Only ripe strawberries are labeled and detected.
2	[28], [37], [54], [68], [77]	Binary classification is typically used (ripe vs. unripe or mature vs. immature). Ripe strawberries are generally described as fully red, smooth, and well-developed, while unripe ones show green or light red coloration and are less developed.
3	[24], [27], [29], [36], [39], [40], [41], [42], [43], [44], [45], [46], [47], [52], [57], [60], [65], [66], [69], [70], [71], [74], [75], [76], [79], [82]	Most adopt a three-stage classification: unripe, partially ripe, and ripe. These stages are typically derived from color progression (green → white → red) or quantified using the red surface ratio (e.g., 0–30%, 30–80%, 80–100%).
4	[25], [38], [61], [72], [80]	Four-class schemes usually follow a detailed maturity progression: green (unripe), white/light red, turning (increasing red coverage), and fully red (ripe). Some define growth stages (fruitlet, expanding, turning, mature) or use red ratio thresholds (e.g., 0–30%, 30–60%, 60–90%, 100%).
5	[26], [62], [64]	Five-class schemes extend maturity grading by introducing subcategories (e.g., fully ripe grade A/B, half ripe grade A/B, unripe, immature, medium, medium well) and categories such as malformed fruits.
6	[32]	Early-Turning (category 0), Green (category 1), Late-Turning (category 2), Red (category 3), Turning (category 4), and White (category 5).

3.4 Data annotation

Data annotation is the process of labeling raw data to create the ground truth required for supervised learning. In the context of computer vision, this involves assigning semantic meaning to image data, for example, by marking object locations, classes, or attributes. The quality and consistency of the annotations are of key importance, as they directly influence model performance and generalizability. Depending on the data and the target task of the DL model, different types of annotations are used, such as BBs for object detection or segmentation masks for pixel-precise identification. For strawberry recognition and ripeness classification, annotations typically include both the fruit's location within the image and class labels indicating its ripeness level. Table 3 shows how the ripeness levels for strawberries were selected in the reviewed studies. Key challenges in data annotation include the precise placement of bounding boxes and handling cases such as partially obscured or visually similar objects. Furthermore, the annotation process is time-consuming and prone to human error.

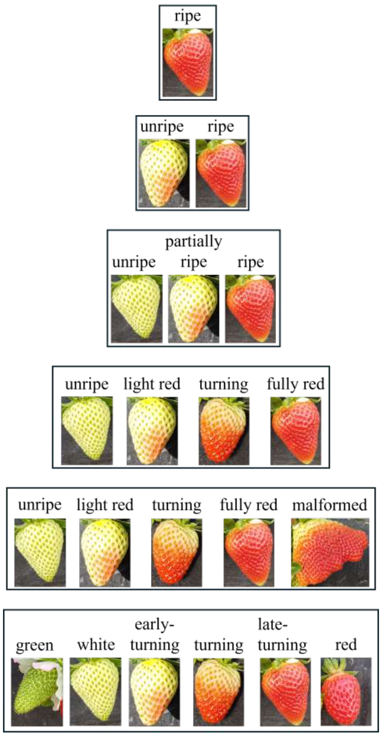

Fig. 2: Example ripeness definitions of reviewed studies. Example images were extracted from the StrawDI_Db1 dataset.

3.5 Data preprocessing and splitting

Before DL models can be effectively trained on raw image data, two fundamental preparation steps are required: preprocessing, which transforms raw images into a format suitable for neural network input, and dataset splitting, which partitions the available data into independent subsets for training, validation, and testing. While preprocessing ensures numerical stability and architectural compatibility, proper splitting is essential for obtaining unbiased estimates of model performance. Figure 3 illustrates a robust end-to-end ML pipeline, from data collection to model evaluation, which serves as a reference framework for this section and establishes the methodological groundwork for the discussion of data augmentation and data leakage in Section 3.6.

3.5.1 Preprocessing

Preprocessing encompasses a set of standard transformations applied to raw images before they are fed into a DL model. Although deep neural networks can learn invariant features to some extent, preprocessing serves three main purposes. Firstly, it normalizes data to a consistent range, thereby improving training stability. Secondly, it ensures uniform input dimensions that are required by most architectures. Thirdly, it reduces irrelevant variability that would otherwise consume model capacity [99].

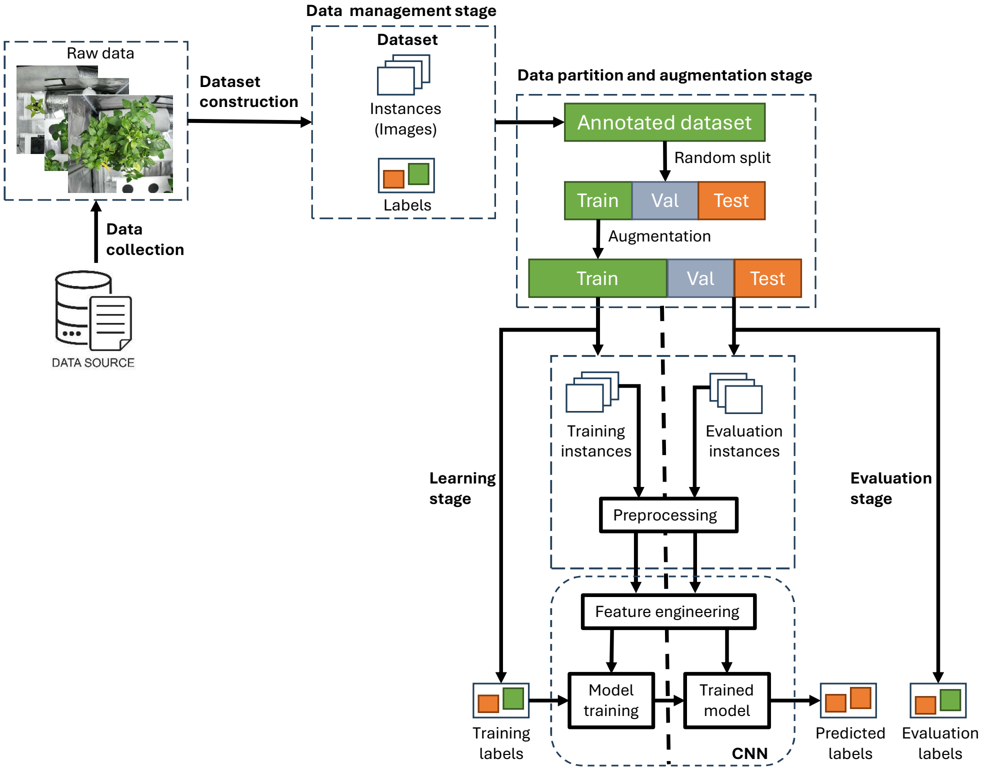

Fig. 3: Schematic overview of a *data leakage-preventing* ML pipeline for supervised object detection. Note the correct sequencing of splitting and augmentation—splitting before augmentation, and the application of augmentation only on training data, either offline or online (on-the-fly).

Preprocessing steps are inconsistently documented across the 60 reviewed studies. Where reported, the following operations are mentioned:

- Resizing: Input images are scaled to a fixed resolution, with 640×640 pixels being the most frequently cited dimension (e.g., [24, 42]). This step ensures compatibility with the input requirements of CNNs such as YOLO variants but may distort aspect ratios, potentially affecting detection performance for elongated objects like strawberries [106].
- Color space conversion: A small number of studies convert RGB images to HSV color spaces to reduce sensitivity to illumination changes, which is particularly relevant in outdoor agricultural environments [64, 94].
- Noise reduction: Occasionally applied (e.g., Gaussian blur, median filtering), though less common in DL pipelines that rely on data augmentation to achieve robustness (e.g., [46, 107]).

Normalization of pixel values, a standard step in many DL pipelines, is rarely mentioned in the reviewed studies, suggesting it is either omitted or treated as implicit. A complete overview of dataset characteristics and augmentation practices is provided in Table 2 (Columns 3 and 4).

3.5.2 Dataset splitting: principles and practices

The second essential step is the partitioning of the available data into disjoint subsets. The standard protocol in supervised DL distinguishes three subsets: a training set for parameter optimization, a validation set for hyperparameter tuning and model selection, and a test set for final unbiased evaluation [99]. To ensure that performance metrics accurately reflect generalization rather than memorization of specific samples, it is imperative that these subsets are mutually exclusive and no image appears in more than one subset.

Among the reviewed studies, the majority report explicit split ratios. Table 2 shows that 8:1:1 is the most frequently used distribution, followed by 7:2:1 and 6:2:2. Only a few studies use a 9:0:1 split (training and test only, no validation set), which is acceptable for smaller datasets but limits hyperparameter tuning capabilities. Stratification, the preservation of class proportions across subsets, prevents scenarios where a rare class (e.g., overripe strawberries) is entirely absent from the test set [106]. Despite the inherent class imbalance in strawberry ripeness stages (see Table 3), this practice is explicitly mentioned in only two studies [46, 47]. Similarly, the random seed used for splitting is reported in fewer than five studies, hampering reproducibility. Furthermore, almost no studies address whether images of the same plant or field row were deliberately separated across subsets, leaving the risk of spatial overlap between training and test sets unaccounted for.

In the agricultural domain, additional considerations apply. Images may contain the same plant from slightly different angles, or multiple images of the same field row. A proper split must ensure that no spatial or temporal overlap exists between subsets. Otherwise, the model may be trained to recognize specific plants rather than general strawberry features [105].

Beyond spatial overlap, a second critical aspect concerns the chronological order of dataset splitting and data augmentation. The correct protocol applies augmentation exclusively to the training set after the split has been performed, referred to as split-then-augment. The validation and test sets remain untouched, preserving them as unbiased references for real-world performance.

Conversely, the flawed augment-then-split protocol applies augmentation to the entire dataset before partitioning. This creates transformed versions of images that may end up in different subsets (e.g., the original image in the training set and a rotated copy in the test set). As a result, the model implicitly learns to recognize augmentation artifacts rather than genuine fruit features, leading to severely inflated performance estimates that do not generalize to real-world scenarios [108, 109].

The following section examines why augmentation is necessary and analyzes the consequences of incorrect sequencing, including data leakage.

3.6 Overfitting, data augmentation, and the risk of data leakage

In ML, overfitting occurs when a model learns not only the underlying patterns in the training data but also noise and spurious correlations that do not generalize to unseen data [99]. Models trained on small datasets inherently fail to generalize well to validation and test data, as they tend to memorize dataset-specific artifacts rather than learning robust features [110].

For strawberry detection tasks, overfitting is particularly problematic due to the inherent variability of agricultural environments. Lighting conditions change throughout the day, fruits appear at different orientations and occlusion levels, and ripeness stages exhibit continuous color transitions. Moreover, publicly available strawberry datasets are constrained in *scene diversity* rather than in raw size. The StrawDI_Db1 dataset, for instance, comprises 3,100 images densely annotated with 17,938 individual strawberry instances [83]. While this constitutes a reasonably large benchmark in terms of the number of object instances, the range of unique environmental conditions, geographic locations, and cultivation systems represented remains limited. These factors are critical for generalization. Consequently, the risk of overfitting to dataset-specific regularities is substantial, and mitigation strategies are required to enable generalization to unseen field conditions.

Data augmentation is a widely adopted technique to mitigate overfitting by artificially expanding the training dataset through transformations that preserve semantic content [106]. The fundamental principle is to increase the diversity of the training set without collecting new data, thereby exposing the model to a broader range of variations that may appear during inference [110].

Furthermore, offline and online augmentation can be distinguished. Offline augmentation applies transformations before training, and the augmented images are saved to disk. Online augmentation, in contrast, refers to transformations applied on the fly during the training process. Online augmentation is generally preferred because it generates unique variations in each training epoch, thereby reducing the risk of overfitting and eliminating the need for additional storage.

Kumar et al. [111] provide a comprehensive taxonomy that distinguishes between basic and advanced augmentation techniques. Basic augmentation includes geometric manipulations (rotation, translation, shearing, flipping), non-geometric manipulations (cropping, noise injection, color space adjustments, jitter, kernel filters), and image erasing techniques (cutout, random erasing, hide-and-seek, grid mask). Advanced augmentation encompasses image mixing methods (MixUp, CutMix, SaliencyMix, among others), AutoAugment techniques, feature augmentation, neural style transfer, and diffusion-based augmentation. [106, 111, 112]

While many studies claim that data augmentation improves generalization and prevents overfitting, e.g., [113, 114], the effectiveness of augmentation depends critically on how and under what conditions it is applied. A foundational analysis by Wong et al. [115] distinguishes between *data warping* (transformations in image space, such as rotation or flipping) and *synthetic oversampling* (generation of new samples in feature space). The authors conclude that data warping is most effective when the underlying data distribution permits continuous transformations. This condition is met in agricultural imaging due to gradual changes in ripeness and smooth variations in lighting.

Empirical evidence from controlled experiments supports this differentiated view. Wang and Perez [110] demonstrated that traditional data augmentation yields substantial improvements for challenging, visually similar classification tasks, while providing only marginal gains for easier tasks. Notably, more complex GAN-based augmentation performed worse than traditional methods, indicating that complexity alone does not guarantee effectiveness. This pattern is reinforced by a recent ablation study by Gao et al. [51], which shows that augmentation can account for the majority of reported mAP gains, whereas architectural modifications alone contribute only marginally. In their study, the proposed architectural changes (SE-MSDWA and CGFM) alone yielded less than 0.1% mAP improvement, while the combination with online Warmup-based augmentation resulted in a gain of 1.94% mAP.

A systematic evaluation by Kumar et al. [111] further confirms that the effectiveness of data augmentation varies substantially across task types. Methods such as MixUp and CutMix, while highly effective for classification tasks, show only moderate benefits for segmentation tasks. Conversely, other techniques, such as ClassMix, are particularly well-suited to semantic segmentation. These domain-specific differences challenge the assumption that data augmentation universally improves generalization across tasks.

Based on these findings, augmentation is most beneficial under the following conditions:

1. The original dataset is small, as reliable data are often rare or hard to obtain [110].
2. The task exhibits high natural variability, including different lighting conditions and object orientations [115].
3. The baseline performance still has substantial room for improvement [110].
4. The visual similarity between classes is high, making the discriminative task challenging [110].
5. The model capacity is high, as deep networks are particularly prone to overfitting when trained on limited data [111].

Conversely, augmentation shows diminishing or no returns on already well-solved tasks (e.g., MNIST digit classification, where Wang and Perez [110] observed no improvement) or when the baseline model already operates near optimal performance. Moreover, excessively aggressive augmentation can lead to underfitting via over-regularization, in which the model becomes unable to capture the underlying patterns in the data [111].

However, as demonstrated in our prior work [105], augmentation alone does not guarantee improved generalization. Beyond the question of which augmentation techniques are most effective, a more fundamental issue concerns the correct application of augmentation itself. If applied incorrectly, it can lead to data leakage, resulting in severely inflated performance estimates that do not reflect the model's true capabilities.

Data leakage refers to the unintentional use of information during model training that would not be available in a real-world deployment scenario [108, 116]. This violates the fundamental assumption that training and test data are independent and identically distributed. A critical and frequently overlooked source of data leakage is *the improper sequencing of data augmentation relative to dataset splitting* [108, 109].

Apicella et al. [109] categorize this as process-induced leakage, where information from the test set inadvertently influences the training phase. When augmentation is applied before splitting, transformed versions of test images, such as rotated copies, may appear in the training set. Consequently, the model learns to recognize augmentation artifacts rather than genuine fruit features, which leads to artificially inflated performance metrics.

The empirical impact of this flaw is substantial. Kumar et al. [111] noted that numerous state-of-the-art studies apply augmentation before dataset splitting without addressing leakage risks, indicating a structural problem in research practice. In our reproducibility study [105], re-evaluating a plant disease detection work using the correct split-then-augment protocol caused reported performance metrics to drop dramatically. This stark contrast demonstrates that reported performance gains may be entirely attributable to data leakage rather than architectural improvements.

Beyond the specific case of augmentation sequencing, Apicella et al. [109] further provided a comprehensive taxonomy of data leakage, distinguishing between data-induced, preprocessing-related, and split-related leakage, and note that such leakage affects studies in medicine, autonomous driving, and agricultural AI.

4 Results: systematic literature analysis

4.1 Evaluation framework

To systematically assess the architectural modifications and their impact on model performance, a structured analysis framework was established. This framework defines three optimization dimensions that reflect the key requirements of real-world agricultural deployment:

- **Detection accuracy:** Evaluated primarily using mean Average Precision at IoU threshold 0.5 (mAP@50, referred to as mAP throughout this review), as this is the most consistently reported metric across the reviewed studies. While some studies report mAP@50:95 or omit the threshold specification, mAP@50 serves as the common denominator for cross-study comparison.
- **Real-time capability:** To assess whether a model meets the requirements of robotic harvesting, detection speed is measured in frames per second (FPS). While no universal FPS threshold exists, practical requirements can be derived from the operational parameters of strawberry harvesting robots. Based on an analysis of robot movement (0.3 m/s with 10 cm fruit spacing), picking cycle duration (0.5–1 s per fruit), and camera field of view (30×30 cm with 50% overlap), a minimum of 30 FPS is required for reliable real-time operation in dynamic greenhouse environments, with demanding scenarios requiring up to 60 FPS [28, 29]. Throughout this review, reported FPS values are compared against this 30 FPS baseline to indicate real-time suitability.
- **Computational complexity:** This is quantified using two complementary metrics: the number of parameters (abbreviated as Params throughout the remainder of this paper) and the number of floating-point operations (GFLOPs). These metrics together indicate the model's size and the computational demand for edge deployment.

Beyond these three evaluation dimensions, architectural modifications were further categorized according to their primary optimization target. This distinction is critical because a modification that reduces model size and computational cost pursues a fundamentally different design objective than one that maximizes detection accuracy, often at the expense of efficiency. Mixing both categories in a single statistical analysis would average opposing effects, potentially masking meaningful patterns or diluting the observed impacts.

Each modification was therefore classified based on the authors' stated primary objective and the reported trade-offs in performance metrics. *Lightweighting-oriented modifications* prioritize the reduction of model parameters, GFLOPs, or inference time, typically targeting deployment on resource-constrained edge devices. For these modifications, moderate accuracy losses (typically below 1–2% mAP) are accepted as a trade-off for improved efficiency. *Accuracy-oriented modifications*, in contrast, prioritize improvements in detection accuracy, recall, or robustness under challenging conditions such as occlusion, small targets, or complex backgrounds. Increases in model complexity or inference time are accepted as a trade-off for higher precision.

For modifications in which the primary optimization target was ambiguous, or in which multiple conflicting modifications were combined without clear ablation, the studies were excluded from the category-specific statistical analysis. Based on this classification, all subsequent analyses were performed separately for lightweighting-oriented and accuracy-oriented modifications.

Using this framework, the performance impact of each modification was quantified as follows: for each architectural modification category (e.g., convolution replacements, feature extractor replacements, attention mechanisms), the impact was measured as the relative change from the respective baseline model reported in each study. Only studies that reported both baseline and modified model performance were included in the quantitative analysis.

The inclusion of a study's reported metrics followed a two-step validation process:

1. **Attribution of improvements:** Only studies that performed ablation experiments or clearly isolated individual modifications were included. Studies with multiple concurrent modifications that could not be unambiguously attributed were excluded from the quantitative analysis. Specifically, [24] (multiple modifications in backbone and neck), [27] (GhostCBS and GhostC3 combined), and [65] (C2f-Faster combined with EMA attention) were excluded because their reported improvements cannot be assigned to a single modification.
2. **Data leakage assessment:** Studies flagged with potential data leakage (see Section 5.2) were excluded from the statistical aggregation for mAP metrics, as their reported gains are likely to be inflated. For parameter counts, GFLOPs, and FPS, however, all studies were retained, as these metrics are architecture-dependent and less susceptible to data leakage artifacts.

For each modification category and optimization target, the statistical analysis considered the minimum and maximum values to capture the full range of observed impacts. These statistics are reported separately for mAP, FPS, GFLOPs, and Params, acknowledging that the number of studies reporting each metric varies considerably across categories.

4.2 YOLO algorithm distribution and evolution

Figure 4 shows the adoption patterns of the YOLO versions, listed in order of frequency, of the 60 studies that proposed YOLO-based architectures.

1. YOLOv8 was the most widely used algorithm in our context, appearing in 28 studies (46.7%; 1 study in 2023, 7 in 2024, 17 in 2025, and 3 in 2026). This is attributed to its extensive community support, balanced accuracy-speed trade-offs, and ease of customization, which facilitate rapid prototyping. A key advantage of YOLOv8 is its anchor-free, center-based approach (with a decoupled head) to object detection. This approach has several advantages over traditional anchor-based methods, such as YOLOv5, YOLOv6, and YOLOv7 [104].
2. Despite its short history, YOLOv11 has been adopted in 13 studies (21.7%; 11 studies in 2025, 2 in 2026) owing to its balanced precision and computational efficiency,

making it suitable for applications ranging from embedded systems to large-scale deployments [117].
3. YOLOv5 appeared in 9 studies (15.0%; 4 studies in 2023, 4 in 2024, 1 in 2025) due to its significant advantages in ease of use and performance, establishing it as a popular choice in the computer vision community [117].
4. YOLOv7 was used in 5 studies (8.3%; 2 studies in 2023, 1 in 2024, and 2 in 2025), despite its recognition for robust feature extraction and fast inference [118].
5. YOLOv9 was employed in only 5 studies (8.3%; 4 studies in 2024, 1 study in 2025) despite its strong balance between accuracy, inference speed, and model size, making it well-suited for real-time embedded or edge deployments [119].
6. YOLOv10 was absent from all included studies on strawberry detection, underscoring its limited adoption within the agricultural community.

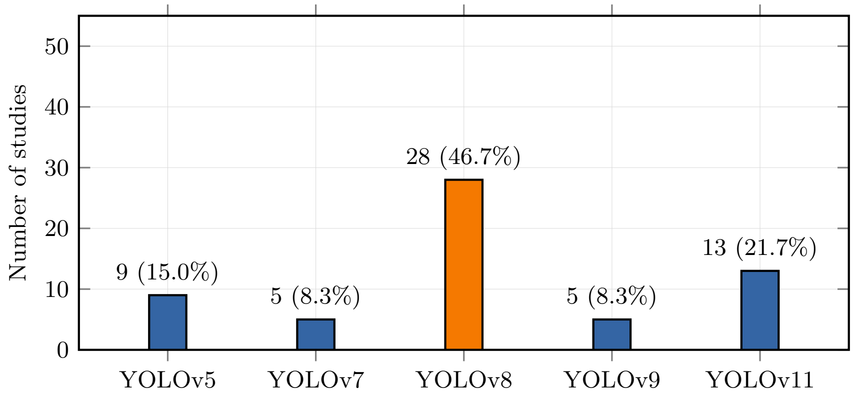

Fig. 4: Distribution of YOLO architectures applied in the reviewed studies (N = 60). Values above bars show absolute counts with percentages in parentheses.

4.3 Architectural modifications and optimization strategies

The reviewed studies not only apply a specific YOLO algorithm to the detection task but also propose improvements to the YOLO architecture under consideration. Similar to the procedure in [21], the improvements can be grouped into three categories: backbone, neck, and head improvements, in addition to pyramid modifications. The improvements introduced in the revised papers are shown in Figure 5. Table 4 summarizes the observed performance impacts for lightweighting-oriented modifications, while Table 5 presents the corresponding statistics for accuracy-oriented modifications.

4.3.1 Pyramid structure modifications

Extending the feature pyramid with an additional 160×160-pixel detection layer (P2) specifically targets small strawberry fruits. Four studies [46, 50, 54, 65] introduced such a layer. The addition of this fourth detection head is consistently accuracy-oriented

Table 4: Lightweighting-oriented architectural modifications and their observed performance impacts. The table reports the observed impact ranges on mAP, FPS, GFLOPs, and parameter count for each modification type. The numbers in parentheses indicate how many modifications provided sufficient data for each metric.

Modification Type	mAP Impact	FPS Impact	GFLOPs Impact	Params Impact	Examples
Backbone Modifications					
Convolution replacements	–0.10% to +1.30% (3)	–	–10.20% to –7.32% (2)	–13.18% to –10.00% (3)	[45, 52, 64]
Feature extractor replacements	–2.0% to +1.30% (14)	–25.77% to +26.23% (7)	–66.78% to +110.00% (15)	–38.32% to +84.92% (16)	[28, 35, 69]
Attention mechanisms	-1.20% (1)	-	-42.41% (1)	-	[27]
Neck Modifications					
Neck architecture	–0.10% to +2.30% (6)	–22.64% to +4.45% (4)	–27.16% to –11.11% (7)	–34.36% to –6.49% (8)	[24, 41, 45]
Convolution replacements	+0.40% (1)	–	–2.27% (1)	–4.74% (1)	[35]
Feature extractor replacements	–0.30% to +1.60% (6)	–9.75% (1)	–24.24% to –3.70% (6)	–27.67% to +22.87% (5)	[34, 35, 81]
Upsampling replacements	–0.20% (1)	–	–6.35% (1)	–5.04% (1)	[91]
Head Modifications					
Other head modifications	–2.60% to +1.00% (6)	+19.36% to +39.67% (2)	–31.71% to –9.72% (6)	–37.76% to –5.29% (6)	[35, 41, 45]
Hybrid Architectures					
CNN–ViT hybrids	–0.20% to +0.60% (2)	+20.74% (1)	–66.78% to –17.19% (2)	–54.05% (1)	[81]

and yields mAP improvements of +0.50% to +2.80%, while GFLOPs increase by up to 60%. In [65], it was shown that the small-target detection head can better extract detailed information from the image, thereby enhancing the interpretability of the data and the portrayal of tiny entities in the strawberry.

Table 5: Accuracy-oriented architectural modifications and their observed performance impacts. The table reports the observed impact ranges on mAP, FPS, GFLOPs, and parameter count for each modification type. The numbers in parentheses indicate how many modifications provided sufficient data for each metric.

Modification Type	mAP Impact	FPS Impact	GFLOPs Impact	Params Impact	Examples
Pyramid Structure					
Pyramid structure	+0.50% to +2.80% (3)	–	+60.00% (1)	–6.40% to +0.76% (2)	[46, 65]
Backbone Modifications					
Convolution replacements	+0.40% to +1.70% (2)	–	+6.52% (1)	+11.90% to +15.58% (2)	[59, 78]
Feature extractor replacements	–0.30% to +2.00% (11)	–55.61% to –0.78% (6)	–6.25% to +35.63% (6)	–24.32% to +31.59% (8)	[29, 49, 74]
Attention mechanisms	+0.20% to +6.69% (11)	–19.89% to +10.26% (8)	–1.19% to +1.23% (5)	–3.70% to +1.95% (9)	[60, 67, 80]
SPPF replacements	+2.30% (1)	–	–	–	[36, 65]
Neck Modifications					
Neck architecture	–0.10% to +2.60% (11)	–52.30% to –1.22% (6)	–3.17% to +71.83% (7)	–25.58% to +56.03% (8)	[26, 51, 77]
Feature extractor replacements	+0.30% to +2.70% (3)	–6.67% (1)	–	–	[54, 81]
Attention mechanisms	–0.35% to +2.63% (7)	–14.05% to +7.69% (3)	–22.22% to +2.47% (4)	0.02% to +2.82% (3)	[28, 53, 61]
Head / Loss Modifications					
Loss function optimizations	–0.60% to +2.60% (17)	–8.55% to +23.08% (10)	No impact (4)	No impact (3)	[31, 61, 65]
Other head modifications	–0.80% to +4.10% (3)	–9.56% (1)	+3.00% to +13.23% (2)	–6.40% to +38.79% (2)	[34, 46, 50]
Hybrid Architectures					
CNN–ViT hybrids	+0.50% to +9.30% (6)	–23.84% to –4.09% (2)	12.22% to +71.83% (2)	–0.12% to +20.58% (4)	[76, 77, 82]

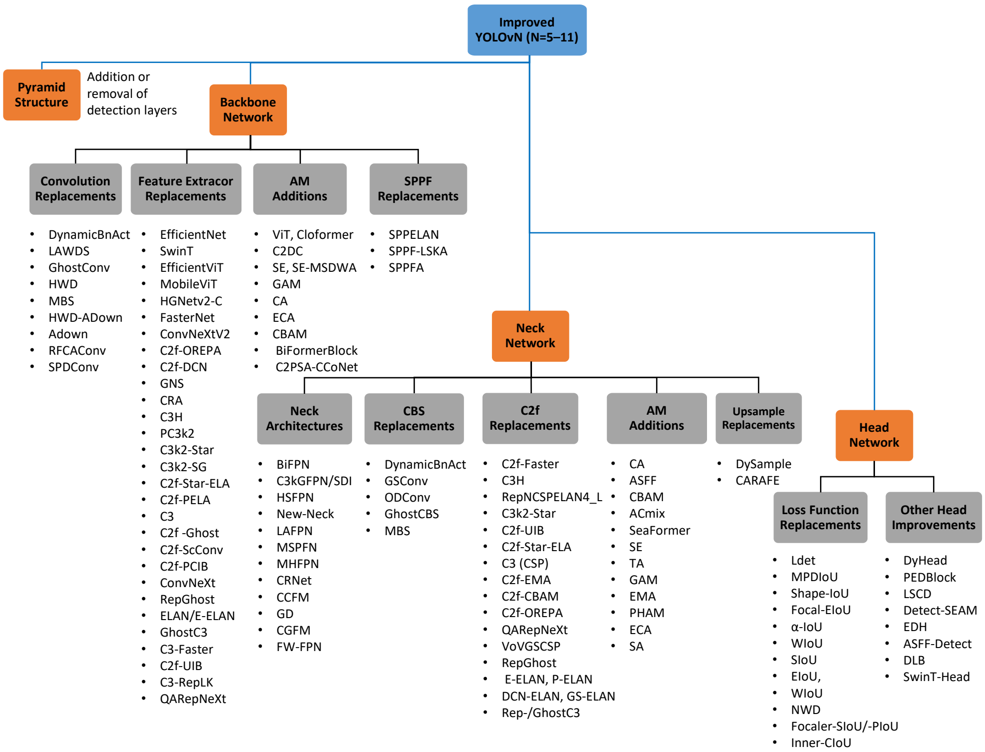

Fig. 5: Taxonomy of the architectural YOLO improvements proposed in the reviewed studies

4.3.2 Backbone improvements

Convolution module replacements

Traditional convolution methods typically employ fixed-size convolution kernels (e.g., 1×1, 3×3, or 5×5) to sample the input feature map. This can result in poor feature extraction for objects with irregular shapes, such as strawberries [31]. Moreover, the computational complexity of traditional convolution modules is a critical factor when deploying models on edge computing platforms with limited computational resources.

Convolution replacements appear in 6 studies (10.0%), with lightweight modules such as HWD and ADown. The classification reveals a clear pattern: lightweighting-oriented replacements achieve parameter reductions of –13.18% to –10.00% and GFLOPs reductions of –10.20% to –7.32%. These methods reduce computational burden through factorization, attention-guided downsampling, and selective application strategies while preserving representational capacity. For instance, Yu et al. [45] replace standard convolutions with ADown modules, achieving a 0.40% mAP gain while reducing parameters by 10.00% and GFLOPs by 7.32%. Similarly, Fu et al. [64] employ LAWDS convolutions, which dynamically generate content-aware adaptive weights to guide downsampling, thereby reducing parameters by 13.18% and achieving a 1.30%

mAP improvement. In contrast, accuracy-oriented replacements (e.g., HWD-ADown in YOLOv9) trade increased parameters (+11.90%) and GFLOPs (+6.52%) for a +0.40% mAP gain. This indicates that not all convolution replacements consistently reduce complexity; some trade increased capacity for marginal gains in accuracy.

Feature extractor replacements

The feature extraction module represents the most frequently modified component, appearing in 32 studies (53.3%). These modifications fall into two complementary categories, each with distinct impacts on model efficiency and accuracy.

Lightweighting-oriented replacements employ modules such as C3-Faster, GhostC3, RepGhost, C2f-Ghost, C3k2-Star, C3k2-PELA, GNS, and StarNet. The most efficient configuration is the complete MobileNet-MCA backbone replacement [35], which achieves a 38.18% reduction in parameters and a 50.00% decrease in GFLOPs. Across 14 methodologically sound studies, mAP changes range from –2.00% to +1.30%, confirming that well-designed lightweight replacements maintain accuracy within 2% of the baseline. A notable exception is the QARepNeXt module [28], which increases parameters by 84.92% and GFLOPs by 110.00% but achieves a 26.23% FPS improvement (from 122 to 145 FPS) through structural reparameterization. This demonstrates that architectural choices can prioritize inference speed over parameter efficiency.

Accuracy-oriented replacements achieve an average mAP improvement of 0.84%, with C2f achieving the greatest improvement of 2.00% [30]. However, this comes at the cost of increased model complexity, as reflected in an average increase of 23.42% in GFLOPs. Furthermore, FPS decreases by an average of 6.32% (range: –55.61% to +26.23% across 13 studies), reflecting the trade-off between accuracy and inference speed.

Novel hybrid designs combine both feature-extraction philosophies by integrating attention mechanisms directly into lightweight feature extractors. Examples include C2f-Star-ELA, C2f-Faster-EMA, and C3k2-PELA, where modules such as ELA or EMA compensate for accuracy losses caused by efficient bottlenecks. A typical example is provided by [48]: By replacing the original YOLOv8n bottleneck with the Star Block, parameters were reduced by 20.13% and GFLOPs by 19.05%. Similarly, Gan et al. [69] demonstrated that embedding ELA attention into a lightweight PConv-based bottleneck (C3k2-PELA) reduces parameters by 7.69% while preserving detection accuracy (+0.50% compared to baseline). These examples illustrate that lightweighting and attention are complementary strategies.

Additions of attention mechanisms

Previously, YOLO models typically relied on either global features or fixed receptive fields for object detection and localization. However, these methods are not ideal for complex scenes, occluded objects, or scale variations, as encountered in crop or disease detection tasks. Incorporating attention mechanisms into the YOLO architecture has emerged as a promising solution to these issues. Later versions of YOLO, such as YOLOv10 and YOLOv11, incorporate attention mechanisms as core components of their baselines, though they use them in strategically different ways. YOLOv10 incorporates attention through its PSA (Partial Self-Attention) module, which captures long-range dependencies using a multi-head self-attention mechanism [120]. YOLOv11 advances this concept with its C2PSA (Cross Stage Partial with Spatial Attention) block, focusing on spatial relationships to improve detection of small or partially obscured objects [121]. YOLOv12 marked a significant shift toward an attention-centric design. It introduced the Area Attention (A^2) module for efficiently processing large receptive fields, as well as Residual Efficient Layer Aggregation Networks (R-ELAN) for enhanced feature aggregation [117]. YOLOv12 also introduced architectural optimizations, including FlashAttention, and adjusted MLP ratios.

Among the reviewed studies, the vast majority of attention mechanism additions are accuracy-oriented. Across 11 such studies reporting reliable mAP values, the average gain is +1.14%, ranging from +0.20% to +6.69%. The impact on parameter count is negligible (average: +0.08%, ranging from –3.70% to +1.95% across 9 studies), and GFLOPs remain nearly unchanged (average: +0.02%, ranging from –1.19% to +1.23% across 5 studies). Inference speed (8 studies) shows a modest average decrease of –3.03%, ranging from –19.89% to +10.26%. A single notable exception is the CBAM module proposed by Tamrakar et al. [27], which pursues a lightweight design objective. While it achieves the largest GFLOPs reduction among all attention mechanisms (–42.41%), this comes at the cost of a 1.20% mAP decrease, illustrating the trade-off between computational efficiency and detection accuracy.

SPPF replacements

Although rare, two of the studies reviewed suggested replacing SPPF with other modules, i.e., SPPF-LSKA in [36] for YOLOv8 and SPPELAN in [65] for YOLOv11. Each of these substitutions led to a 2.30% increase in mAP, which is remarkable given that improvements to the SPPF module have largely gone unnoticed until now.

4.3.3 Neck improvements

Neck architecture modifications

Although the PAN-FPN, as the standard neck structure in YOLO algorithms, can perform multi-scale feature fusion, it is ineffective at suppressing redundant information. This leads to the accumulation of background interference, limiting its ability to discriminate in complex scenarios [71]. Therefore, modifying the neck network can be beneficial for agricultural applications.

Neck modifications appear in 21 studies (35.0%), including the Bidirectional Feature Pyramid Network (BiFPN, used in 5 studies) and High-Level Screening Feature Fusion Pyramid Network (HS-FPN, used in 2 studies) as the most frequent variants. BiFPN optimizes the topology by pruning single-input edge nodes, which have limited utility for feature fusion because they cannot integrate multi-source information [45]. The HSFPN reduces information loss during transmission by creating more efficient channels for information to flow between levels, thereby maintaining a balance between high-level semantic and low-level spatial information throughout the network [36]. The impact of neck modifications varies considerably depending on the design objective.

Lightweight-oriented modifications, such as the MS-FPN in YOLOv8 [41] and CCFM in YOLOv8 [48], achieve parameter reductions of 30.00% and 31.82%, respectively, with GFLOPs reductions of 27.16% and 19.05%, at the cost of minor mAP changes (–0.10% and +1.30%).

Conversely, accuracy-oriented modifications (e.g., C3kGFPN-S [80], GD [52], CGFM [51]) show the opposite trend. Here, average mAP gains of 0.99% across 11 studies can be achieved at the cost of parameter increases up to 56.03%, and GFLOPs increases up to 71.83%. The extreme case is the CGFM module [51], which increases parameters by 56.03% and reduces FPS by 52.30% for a modest 0.88% mAP improvement, illustrating the substantial trade-offs that certain architectural choices entail.

Convolution module replacements

Standard convolution operations in the neck network introduce substantial computational redundancy, hindering edge deployment. Lightweight convolution modules address this limitation through factorization and selective application strategies. However, convolution replacements specifically within the neck remain rare in the reviewed literature. Only one study, proposed by Yang et al. [122], replaces standard downsampling convolutions in the neck with the FocusDownNet (FDN) module. FDN employs adaptive attention-guided downsampling, reducing parameters by 4.74% and GFLOPs by 2.27%. The scarcity of such modifications suggests that most research focuses on backbone convolutions, leaving neck-specific convolution optimization underexplored.

Feature extractor replacements

The feature extraction module in the neck network plays a critical role in aggregating multi-scale semantic information prior to detection. Replacing standard feature extractors with specialized blocks can enhance feature fusion efficiency and reduce computational overhead. Neck-specific feature extractor replacements appear in 11 studies (18.3%) and fall into two distinct categories.

Accuracy-oriented replacements (4 studies) aim to improve detection performance while maintaining or slightly reducing model complexity. Across these studies, mAP gains range from +0.30% to +2.70% (average: +1.40%). The GS-ELAN module [34] achieves the highest gain (+1.60%) while reducing parameters by 22.87% and GFLOPs by 7.71%. Only one study evaluates the impact on FPS, with a decrease of 6.67% [33].

Lightweighting-oriented replacements prioritize parameter and GFLOPs reductions at the cost of minor accuracy losses. mAP changes range from –0.30% to +1.60% (average: +0.60%). The E-ELAN module [122] achieves substantial reductions (–27.67% parameters, –24.24% GFLOPs) with negligible mAP loss (–0.10%). Reported GFLOPs across 6 studies range from –24.24% to –3.70% (average: –11.44%). Apart from a single study [123] that reported a –9.75% drop in FPS, no FPS data were provided for lightweighting-oriented modifications.

Upsampling replacements

Standard upsampling operations (e.g., nearest-neighbor or bilinear interpolation) can introduce aliasing artifacts and fail to preserve fine-grained spatial details crucial for small-fruit detection. Advanced upsampling modules replace these with content-aware mechanisms that learn adaptive sampling strategies. Upsampling replacements appear in 4 studies (6.7%), including CARAFE in YOLOv5 [26] and DySample in YOLOv8 [44], YOLOv9 [62], and YOLOv11 [91]. These modules learn feature-dependent reassembly kernels to better preserve edge details and small-target structures.

However, the reported mAP gains in two studies are compromised by data leakage (see Section 5.2). Excluding these, only the evaluation of DySample [91] remains methodologically sound, showing a marginal mAP decrease of 0.20% while achieving GFLOPs and parameter reductions of 6.35% and 5.04%, respectively. This suggests that while upsampling replacements can improve computational efficiency, their impact on detection accuracy in strawberry fruit detection remains inconclusive based on the current literature.

Additions of attention mechanisms

Attention mechanisms enhance the neck's ability to suppress background interference and emphasize fruit-relevant features during multi-scale fusion. Incorporating attention modules into the neck network selectively recalibrates feature responses across channels and spatial positions, improving detection accuracy for occluded or densely packed strawberries. Attention additions in the neck appear in 8 studies (13.3%), including triplet attention (TA) in YOLOv5 [28], CBAM in YOLOv8 [53, 124] and YOLOv11 [65], iRMB in [62] and ACmix in YOLOv9 [61]. These modules operate on fused feature maps to prioritize informative regions while suppressing redundant background signals.

According to the analysis of these attention mechanisms, the average percentage change in accuracy (mAP) ranged from -0.35% to +2.63%, with a average of +1.12%. The highest gain is achieved by triplet attention [28], albeit with a 5.84% FPS decrease, illustrating the trade-off between accuracy and inference speed. Conversely, ACmix [61] improves FPS by 7.69% with a modest 0.58% mAP gain, demonstrating that certain attention modules can enhance efficiency.

Regarding computational overhead, attention mechanisms in the neck introduced only marginal increases in parameters and GFLOPs across the reviewed studies. The highest parameter increase was reported for CBAM in a hybrid YOLOv8 architecture, adding 2.82% [124], while the highest GFLOPs increase was 2.47% for TA in YOLOv8 [85]. On average, attention mechanisms increased parameters by 0.95% and decreased GFLOPs by 4.25% (the latter reflecting a slight decrease due to the iRMB module in YOLOv9 [125], which reduced GFLOPs by 22.22%). These figures indicate that attention mechanisms introduce negligible to modest computational overhead in the neck.

4.3.4 Head improvements

Loss function optimization

GIoU, DIoU, and CIoU are the default bounding-box regression loss functions in YOLO algorithms. Enhanced YOLO algorithms improve bounding box localization by modifying the loss function to jointly account for angle, distance, shape, and IoU.

Loss function modifications appear in 22 studies (36.7%), with WIoU (5 studies), (Focal)-SIoU (4 studies), and (Focal)-EIoU (3 studies), emerging as the most frequently considered alternatives to default losses, alongside variants such as Inner-CIoU and Focal-CE. These advanced IoU variants account for geometric factors such as distance, aspect ratio, and angle to provide more informative gradients during regression. All modifications are accuracy-oriented, as they do not alter network architecture.

The observed average mAP improvement is +0.76%, with a range of –0.60% to +2.60%. Notably, Huang et al. [29] trade a 0.30% mAP decrease for a 2.09% FPS gain, illustrating that loss-function modifications can be tailored to prioritize inference in real-time harvesting applications such as picking-point detection. Since loss function modifications do not alter network architecture, they have no impact on model parameters or GFLOPs.

Other head modifications

Beyond loss function optimization, architectural modifications to the detection head directly influence bounding box regression quality and classification confidence. Head modifications appear in 10 studies (16.7%).

Lightweighting-oriented variants (DLB, LSCD, PEDBlock) achieve parameter reductions of –37.76% to –5.29% (average: –22.77% across 6 studies) and GFLOPs reductions of –31.71% to –9.72% (average: –18.88% across 6 studies), but mAP changes range from –2.60% to +1.00% (average: –0.22%). The PEDBlock [45] offers the strongest efficiency gains (–33.33% params, –31.71% GFLOPs), albeit with a 2.60% mAP loss. This is a trade-off that may be acceptable for extreme lightweighting scenarios.

Accuracy-oriented head modifications (SwinT-Head, Head, DyHead) achieve mAP changes of –0.80% to +4.10% (average: +1.70%) but increase parameters by +38.79% and GFLOPs by +13.23% in the extreme case [34]. These modifications redesign the detection head to improve localization precision for occluded or densely packed strawberries, often at the cost of computational efficiency.

4.4 Hybrid architectures (CNNs+ViTs)

Overall, YOLO-based detectors remain the preferred choice for real-time applications due to their lightweight architecture. Meanwhile, transformer-based models are gaining traction in precision agriculture for their improved accuracy in complex scenarios [119]. ViTs [126] have emerged as an alternative to convolutional architectures by relying on self-attention mechanisms for feature extraction. In contrast to CNN-based detectors, ViTs model long-range dependencies more effectively, which is particularly advantageous in complex agricultural environments characterized by occlusions, varying illumination, and dense object distributions. These hybrid approaches aim to balance the local feature-extraction strengths of CNNs with the global contextual modeling capabilities of Transformers [127].

In this review, we define a model as a hybrid CNN-ViT if it integrates at least one Vision Transformer component (e.g., Swin Transformer block, multi-head self-attention, or DETR-style encoder) into a predominantly CNN-based architecture. This includes replacing the entire backbone with a ViT while keeping the CNN-based neck and head; inserting ViT blocks into specific stages of the backbone or neck; and using attention modules that rely on ViT-style global self-attention rather than CNN-based local attention (e.g., CBAM, SE). Purely attention-based mechanisms operating on CNN feature maps (e.g., SE, CBAM) are not considered hybrid CNN-ViT models.

8 studies proposed hybrid CNN–ViT architectures. With the exception of 2, all of these are focused on accuracy, as the integration of Transformer components increases model capacity. Across methodologically sound studies, mAP gains range from +0.50% to +9.30% (average: +2.45%). However, it should be noted that these gains are accompanied by substantial costs, as evidenced by the significant increase in GFLOPs of 71.83% (as reported in [77]) and parameters ranging from –0.12% to +20.58% (with an average of +12.09% across 4 studies).

Lightweight-oriented hybrid modifications were reported in only two studies. Wu et al. [81] integrated an EfficientViT into the YOLOv8 backbone, achieving a substantial GFLOPs reduction of 66.78% at the cost of a marginal mAP decrease of 0.20%. Xu et al. [68] proposed MVYOLO-Tiny, a YOLOv11-based architecture incorporating MobileViTv2, which reduced GFLOPs by 17.19% and parameters by 54.05% while improving mAP by 0.60%.

Overall, ViT-based approaches demonstrate strong capabilities in modeling complex visual relationships and handling challenging scenarios such as occlusion and variable lighting. However, compared to CNN- and YOLO-based detectors, they are typically associated with higher computational requirements and lower inference speeds. While the observed mAP gains in strawberry detection are modest (+0.50% to +1.70% for most hybrid integrations, with one outlier achieving +9.30% [76]), other ViT variants have shown greater gains on related tasks [128], suggesting potential for further optimization. The limited number of studies and moderate performance gains suggest that ViTs are still in an early adoption phase within strawberry detection tasks. Further research is necessary to ascertain the potential of this approach.

4.5 Performance analysis

On average, YOLO-based architectures achieved a mAP of 92.09% across all 60 reviewed studies, representing a gain over the corresponding baseline of 2.66%. Disaggregated by version, the highest average gains were achieved for YOLOv5 (+3.78%) and YOLOv11 (+2.80%), while YOLOv9 showed the smallest improvement (+1.93%). Based on the systematic categorization of architectural modifications, two fundamental design philosophies emerge across the reviewed literature: *lightweighting-oriented* (34.0% of studies) and *accuracy-oriented* (66.0% of studies). This section synthesizes the observed performance impacts across all modification categories to identify broader patterns and derives practical recommendations based on the data in Table 5.

Overall trends across all modifications

Aggregating all 49 lightweighting-oriented modifications reveals consistent efficiency gains with minimal impact on accuracy. On average, lightweighting modifications reduce parameters by 17.05% and GFLOPs by 16.48%, while achieving a slight 0.24% increase in mAP and a 4.83% improvement in FPS. These results confirm that lightweighting modifications consistently improve computational efficiency without significantly compromising detection accuracy.

In contrast, 92 individual accuracy-oriented modifications prioritize performance gains at the expense of computational efficiency. The average mAP improvement is +1.16%, while parameters increase by 5.17% and GFLOPs increase by 10.87%. FPS decreases by 7.67%, reflecting the trade-off between accuracy and inference speed.

Category-specific insights

Synthesizing the results from the previous subsections, four categories of modifications stand out in terms of frequency or impact.

Feature extractor replacements in the backbone represent the most frequently modified component (53.3% of studies). Lightweighting-oriented variants achieve parameter reductions of –12.61% and GFLOPs reductions of –15.00% on average, with minimal mAP change (+0.05%). Accuracy-oriented variants deliver average mAP gains of +0.84% but increase GFLOPs by 12.51% and decrease detection speed by 17.39%.

Attention mechanisms in the backbone offer the most favorable trade-off. Across 17 accuracy-oriented studies, the average mAP gain is +1.04% with negligible parameter overhead (average +0.08%) and only modest FPS reductions (average –3.03%). Crucially, certain attention modules (e.g., CGA in YOLOv9) can even improve inference speed (+10.26%). Attention should therefore be considered a default addition for accuracy-oriented applications.

Loss function optimizations provide zero-cost accuracy improvements (average: +0.76% mAP) without affecting parameters or GFLOPs, making them universally beneficial. Furthermore, FPS changes are minimal (average +2.54%).

Hybrid CNN–ViT architectures with an accuracy orientation show comparatively high gains in accuracy (average: +2.45% mAP) but at prohibitive computational cost (average +12.09% parameters). Lightweight-oriented hybrids were reported in only two studies. Lightweight-oriented hybrids, reported in only two studies, achieved substantial efficiency gains: one reduced GFLOPs by 66.8% at a marginal accuracy loss of 0.2% mAP, while the other cut parameters by 54.1% and GFLOPs by 17.2% while actually improving mAP by 0.6%. These results suggest that ViT integration can be viable for edge deployment, with MobileViT-based designs offering a favorable trade-off between accuracy and efficiency.

Four main conclusions emerge from this analysis: For lightweighting priorities, replacing standard feature extractors with lightweight variants (e.g., C2f-Ghost, C3k2-Star) yields average parameter reductions of 12.61% without accuracy loss, and combining lightweight bottlenecks with embedded attention (e.g., C2f-Star-ELA) maximizes efficiency. For accuracy priorities, adding attention mechanisms to the backbone is the most efficient modification, delivering average mAP gains exceeding 1% at negligible computational cost. Loss function optimizations should always be considered, as they improve mAP by approximately 0.76% with no additional computational overhead. Finally, CNN–ViT hybrids should be approached with caution: while accuracy gains are promising, current implementations are too computationally expensive for real-time edge deployment, though lightweight ViT integration represents a promising research direction.

Synergistic combinations

Beyond individual modifications, several studies demonstrate that combining complementary techniques yields superior overall performance. For instance, Gan et al. [69] combine a lightweight C3k2-PELA bottleneck with an LA-FPN neck, achieving parameter and GFLOPs reductions of 11.50% and 7.90%, respectively, while improving mAP by 2.50% at a modest FPS decrease of 4.70%. Similarly, Liu et al. [65] integrate a lightweight C2f-Ghost backbone with EMA attention and an additional P2 detection layer, achieving a 2.70% mAP gain at substantially reduced computational cost (–36.80% parameters, –34.70% GFLOPs) and a 12.80% FPS improvement. These examples illustrate that lightweighting and accuracy-oriented modifications are not mutually exclusive but can be strategically combined for synergistic effects.

5 Limitations, challenges, and future research directions

Despite the substantial progress outlined in the previous sections, research on DL-based strawberry fruit and ripeness detection still faces several fundamental limitations and unresolved challenges. These issues affect the entire research pipeline, including data acquisition, preprocessing, model architectures, experimental design, and evaluation. In particular, limitations in dataset availability and diversity, methodological shortcomings such as potential data leakage, and the lack of standardized benchmarks significantly constrain the reliability and comparability of reported results. In addition, emerging model paradigms, such as Transformer-based architectures, introduce new challenges related to data requirements and generalization. The following subsections analyze these aspects and highlight key challenges that must be addressed to enable more robust, reproducible, and application-oriented research in this domain.

5.1 Data availability and diversity

As shown in Table 2, several datasets are already being used in research. Nevertheless, many new datasets are still being created, though not all are publicly available. One particular problem is that there is currently no uniform standard specifying how such data should be collected and what requirements they should fulfill. The available datasets have their own strengths and weaknesses, with limited diversity being one of the biggest weaknesses. Each dataset is generally limited to the specific environmental conditions under which the data was collected. Combining multiple datasets could partially address this problem, but would require intensive manual processing. This includes not only removing unsuitable image data (e.g., images that are too dim or overexposed, augmented data, or images with watermarks) but also removing duplicates and highly similar images.

To ensure that the model generalizes well, the dataset must be diverse not only in environmental conditions but also in relevant features, such as brightness, contrast, color distribution, fruit size, and fruit position.

Currently, there is no established standard for quantitatively assessing dataset diversity. Initial approaches, such as the Vendi-Score, aim to fill this gap. This metric measures the intrinsic diversity of a dataset by considering only the data present, without requiring a reference distribution [129]. The Vendi-Score thus assesses the overall diversity of a dataset but does not provide information on the distribution within specific feature categories or on which categories are underrepresented or completely missing.

Current research, therefore, focuses on developing methods that analyze datasets with respect to features relevant to object recognition. The goal is to better quantify diversity, for example, to use data augmentation in a more targeted and effective manner.

5.2 Alarming rate of potential data leakage

A critical issue identified across the reviewed studies is the high risk of unintended data leakage, which can lead to overly optimistic performance estimates and limit the validity of reported results. Data leakage occurs when information from the training set is unintentionally shared with the validation or test set. In agricultural image datasets, this risk is particularly pronounced due to the common use of data augmentation and sequential image acquisition. Many studies generate augmented samples prior to splitting the dataset. As a result, visually similar or nearly identical images may appear in both training and evaluation sets, artificially inflating performance metrics [105, 108, 109].

To better assess the extent of this issue, we systematically analyzed the 60 reviewed studies with respect to their augmentation and dataset splitting protocols.

Although 45 studies (75.0%) reported using data augmentation, indicating a general awareness of overfitting, our analysis revealed substantial methodological inconsistencies. In particular, 20 studies (33.3%) either explicitly followed or strongly implied an augment-then-split protocol, in which data augmentation is applied before the dataset is split into training and test sets. In contrast, only 23 studies (38.3%) clearly adhered to the recommended split-then-augment procedure, and just 6 studies (10.0%) reported using online augmentation during training, which inherently avoids this issue. The remaining 15 studies (25.0%) did not report using any augmentation.

Overall, these findings indicate that in exactly one-third of studies, reported performance metrics may be affected by data leakage. This observation is consistent with broader concerns raised in precision agriculture research and highlights a challenge in current experimental practices [105, 130].

Beyond the need for stricter experimental protocols, several research directions emerge from our findings. There is a lack of a principled understanding of when and how data augmentation should be applied. Given that a substantial proportion of studies employ augmentation in a potentially flawed manner, future research should not only focus on enforcing correct pipelines but also on critically re-evaluating the role of augmentation itself.

More importantly, there is a clear need to move beyond the current largely heuristic and insufficiently validated use of augmentation toward evidence-based guidelines. Future work should therefore investigate under which conditions augmentation provides measurable benefits and how its effectiveness depends on dataset characteristics (e.g., size, diversity, class imbalance). In this context, alternative strategies beyond traditional geometric and photometric transformations should be considered. For example, feature-level approaches such as SMOTE (Synthetic Minority Oversampling Technique) generate synthetic samples in the feature space rather than modifying raw images, offering a fundamentally different way to address class imbalance. Such approaches have already shown promising results in related agricultural tasks, including cassava disease classification and weed detection [131, 132]. Exploring these methods in the context of strawberry detection may provide more controlled and potentially less leakage-prone alternatives to conventional augmentation pipelines.

5.3 Lack of standardized benchmarks and limited result comparability

A fundamental limitation of the reviewed literature is the lack of direct comparability between reported results. While Table 2 summarizes performance metrics across multiple studies, these results are obtained on different datasets and under varying experimental conditions. Consequently, no fair or direct comparison of model performance is possible within the scope of this review.

This lack of comparability is primarily driven by the heterogeneity of datasets used in current research. The reviewed studies rely on a wide range of data sources, including public datasets, self-built datasets, and combinations thereof, each characterized by different environmental conditions, acquisition setups, and levels of difficulty. Performance metrics such as mAP are strongly influenced by these dataset-specific properties, making it difficult to attribute performance differences to the underlying models.

To partially mitigate this issue, this review focuses on relative performance improvements (e.g., mAP gain) rather than absolute metric values, as they provide a more robust indication of model improvements within the same experimental setup. However, even such relative measures remain limited in their comparability across studies, as they are still inherently tied to dataset-specific characteristics.

The lack of standardization is further exacerbated by inconsistencies in dataset splitting and experimental protocols. Our analysis shows that the majority of studies (53 out of 60, or 88.3%) rely on self-built or combined datasets without clearly defined or reproducible split strategies. Even when public datasets with official train/validation/test partitions are available, many studies do not explicitly state whether these splits are used or modified. This lack of transparency not only hinders reproducibility but also complicates the interpretation and comparison of reported results.

Compounding this issue, the reviewed studies exhibit considerable heterogeneity in which metrics are reported. While mAP@0.5 is nearly ubiquitous, other metrics such as F1-score are frequently omitted. For computational efficiency, some studies report model size in MB rather than parameter count, or inference time in ms rather than FPS. These inconsistencies further impede cross-study comparability.

As discussed in Section 5.2, inconsistent handling of preprocessing and augmentation steps can further distort performance estimates. As a consequence, current state-of-the-art comparisons must be interpreted with caution. Apparent performance improvements may be due in part or entirely to differences in dataset characteristics or experimental setups rather than to genuine methodological advances.

However, even with standardized benchmarks in place, comparability is undermined by insufficient methodological transparency within the reviewed studies. Critical evaluation parameters that directly influence mAP, such as confidence thresholds, non-maximum suppression (NMS), IoU thresholds and precision-recall curve interpolation methods, are rarely specified. Without these details, it is impossible to independently verify reported metrics, even when the same dataset is used. This issue is further exacerbated by the fact that only a minority of the reviewed works include ablation studies that isolate the effect of each proposed architectural modification. More critically, only one of the studies we examined [51] quantifies the specific impact of data augmentation on final performance in their ablation experiments, revealing that augmentation accounted for the majority of the reported mAP gain. This omission is particularly glaring given the data leakage risks documented in Section 5.2. Consequently, many of the marginal mAP improvements reported in the literature (typically 1–3%, see Table 2) cannot be reliably attributed to architectural innovations rather than to undocumented differences in training configurations, evaluation protocols, or dataset-specific biases.

5.4 Inductive bias and data hunger of Transformer models

As revealed in most of the reviewed studies, CNN-based models (R-CNNs and advanced versions, as well as variants of YOLO) typically achieved modest gains, i.e., $\leq 5\%$ over the baselines. Studies reporting higher gains, such as [24, 36], may be overly optimistic due to potential data leakage. Moreover, they teach us that remarkable modifications to architectures are required rather than simply introducing or replacing some modules. Examples of powerful modifications include replacements of the entire backbone (e.g., EfficientNetV2 in [42] and StarNet-ELA in [48]) or of the neck structure (e.g., FW-FPN in [24] and CCFM in [48]).

Unlike CNNs, which make assumptions such as locality (pixels close together are related) and translation equivariance (objects do not change based on position), ViTs treat images as sequences of patches. This enables them to model long-range, global dependencies from the outset. However, ViTs require large, high-quality datasets to perform well, making them data-hungry. This reliance on vast amounts of data can pose a significant challenge in agriculture, where acquiring labeled datasets is costly and time-consuming. As a result, ViTs may struggle to achieve optimal performance without substantial data augmentation or pretraining on large-scale datasets, such as ImageNet. [127]

Among the reviewed studies, only three applied DETR-Transformers, one of the first types of Transformers introduced to surpass the Faster R-CNN baselines. In the studies [77, 86, 133], mAP gains under 3% were achieved. However, there are no studies that apply later generations of Transformers, such as PVT and BiFormer, to strawberry detection.

One way to address the lack of inductive biases in ViTs is to use a CNN to extract low-level features and feed the resulting representation into a ViT. This approach combines the spatial inductive biases of CNNs with the global modeling capabilities of ViTs. Introducing a few conventional inductive biases in the early stages of ViT processing balances inductive bias with the learning capacity of Transformer blocks while influencing the optimization behavior of ViTs [127].

Indeed, hybrid models that combine CNN backbones to extract low-level features with Transformers for high-level global modeling can yield larger mAP increases, as exemplified by studies [76, 133]. Although not yet observed in more reviewed studies, this underscores the superior potential of ViT-based models for accuracy-critical tasks such as strawberry ripeness detection. Gerz et al. [128] have recently demonstrated in a plant disease detection study on the PlantDoc dataset that integrating specific ViT backbones into YOLOv8 can elevate the performance substantially. This approach can be applied to strawberry detection in future work.

5.5 Multi-task framework for autonomous harvesting robots

The development of reliable strawberry harvesting robots requires not only accurate detection of fruit and ripeness but also the integration of several interdependent processing steps into a seamless pipeline. A complete multi-task framework must address four key tasks: fruit detection in complex field environments, ripeness classification, three-dimensional spatial pose estimation, and identification of a suitable picking point that enables damage-free stem separation.

Despite significant progress in object detection and segmentation [95, 134], no robust end-to-end pipeline currently exists that reliably combines all these steps under realistic field conditions. As shown in Table 2, the 60 reviewed studies focus on fruit detection and ripeness classification, while only 13 studies address picking point detection, and none provide a fully integrated solution. Specifically, two major gaps persist. First, although attention mechanisms have been shown to improve occluded fruit detection (see Section 4.4), as in [66]. Only a few of the reviewed studies [31, 76] explicitly handle detection based on occlusion levels for harvesting decisions. Second, only two studies [48, 71] incorporate depth information from RGB-D cameras, and none evaluate 3D localization accuracy across varying levels of occlusion.

Addressing these gaps requires moving beyond isolated detection tasks toward modular, integrated, multi-task frameworks. Several research directions appear particularly promising. End-to-end differentiable pipelines, which jointly optimize detection, pose estimation, and 3D localization, have recently demonstrated improved real-world performance in other domains such as robotic grasping in cluttered environments [135] or multi-task picking operations [136]. Transferring this paradigm to strawberry harvesting could significantly reduce error accumulation across processing stages. Furthermore, occlusion-aware picking point selection could benefit from hybrid CNN-ViT models (see Section 4.4), whose ability to model global fruit-stem relationships may improve localization accuracy for heavily occluded fruits. Finally, while YOLOv11 achieves over 200 FPS on high-end GPUs (see Table 2), deployment on embedded systems such as the NVIDIA Jetson series requires further model compression and quantization, as discussed by Sapkota et al. [117]. The focus of future research should therefore shift from isolated detection accuracy to holistic, deployment-ready multi-task frameworks that combine deep learning, geometric analysis, and predictive models under real-time constraints.

6 Conclusions

This systematic review analyzed 60 DL-based studies on strawberry fruit and ripeness detection published between 2023 and 2026. The following main conclusions can be drawn based on the analysis of data collection and preprocessing strategies, architectural choices, performance evaluation methods, and experimental setups.

Data-related findings

The majority of studies (53 of 60) rely on self-built datasets, with the StrawDI_Db1 dataset being the most frequently used public benchmark (8 studies). However, dataset diversity remains a critical limitation, as most collections are constrained to specific environmental conditions, cultivation systems, and geographic locations. A more alarming finding concerns data augmentation practices: 33.3% of the reviewed studies either explicitly followed or strongly implied an augment-then-split protocol. This poses a significant risk of data leakage, resulting in overly optimistic performance estimates. None of these papers mentioned or addressed leakage control, e.g., by checking and removing duplicates between the training and evaluation sets.

Only 38.3% of the studies adhered to the recommended split-then-augment procedure. This methodological flaw is not inherent to the agricultural domain. As demonstrated by Sireesha et al. [137] in their work on citrus disease classification, rigorous protocols, namely freezing train/validation/test splits before applying any augmentation, are feasible and effectively prevent data leakage even with small, challenging datasets. Their explicit statement that augmentation was applied "strictly enforced following splitting" serves as a best-practice benchmark that should be adopted across the strawberry detection literature. Some of the studies examined—albeit very few—exemplarily and explicitly addressed the potential problem of data leakage, such as [41, 69]. In the future, researchers should include the effect of data augmentation in their ablation studies, as discussed in the paper [40].

Architectural findings

Among the reviewed papers, YOLOv8 and its improved variants dominate the field (46.7%), followed by YOLOv11 (21.7%) and YOLOv5 (15.0%). Hybrid CNN-ViT architectures have recently gained traction, with 8 studies demonstrating their potential to model long-range dependencies and handle occlusions. However, their adoption remains limited due to higher computational demands and the need for large-scale pretraining.

Performance and reproducibility findings

Most architectural improvements yield moderate performance gains, typically 1.87% $\pm$2% mAP improvement over baselines. Studies reporting substantially higher gains ($> 5\%$) are frequently associated with potential data leakage or insufficiently documented experimental protocols. Direct comparability across studies is severely limited by heterogeneity in datasets and evaluation protocols, as well as by the lack of standardized benchmarks. Additionally, incomplete reporting of hyperparameters, framework versions, and hardware configurations limits reproducibility.

Practical implications

The development of autonomous strawberry harvesting systems requires not only accurate fruit detection and ripeness classification but also precise localization to determine picking points and enable collision-free navigation. Current research focuses predominantly on detection accuracy, while downstream tasks such as 3D-pose estimation and real-time inference on edge devices remain underexplored. YOLOv8 and YOLOv11 currently offer the best trade-off between accuracy and inference speed for real-time applications, whereas hybrid CNN-ViT models may be preferred for accuracy-critical tasks when computational resources permit.

Limitations of this review

This review is limited to peer-reviewed articles and conference papers published in English between 2023 and May 2026, indexed in Scopus, IEEE Xplore and Google Scholar. Preprints, non-English publications, and studies focusing primarily on robotics or sensor development were excluded. Furthermore, the lack of standardized benchmarks prevented quantitative meta-analyses or direct performance comparisons across studies. Wherever possible, we focused on relative performance gains rather than absolute metric values to mitigate this limitation.

In summary, DL, particularly YOLO-based architectures, has substantially advanced strawberry fruit and ripeness detection. However, methodological issues such as data leakage, lack of standardized benchmarks, and insufficient reporting of experimental details currently limit the reliability and comparability of reported results. Addressing these challenges through rigorous protocols, diverse datasets, and reproducible benchmarking will be essential for translating research progress into robust, real-world agricultural applications, including autonomous harvesting systems.

Statements and Declarations.

Funding. This research was partly funded by the Central Innovation Programme for small- and medium-sized enterprises (SMEs) of the German Federal Ministry for Economic Affairs and Climate Action, grant number ZIM/KK5149913JF4/Erntbot.

Competing interests. The authors declare no conflict of interest.

Ethics approval. Not applicable.

Consent to participate. Not applicable.

Consent to publish. Not applicable.

Data availability. The sources of the datasets mentioned in this review are cited within the manuscript. No new data were created or analyzed in this study.

Author contributions. M.J.: Conceptualization, methodology, supervision, writing—DL architectures (original draft, review & editing). F.G.: Conceptualization, visualization, investigation, formal analysis, writing—data leakage and DL computation (original draft, review & editing). O.-T.A.: Conceptualization, visualization, investigation, formal analysis, writing—data collection and preprocessing (original draft). L.A.-S.: Funding acquisition, supervision, resource management, writing—review & editing. All authors reviewed and approved the final manuscript.

References

[1] Adel, M.: Strawberries in the Global Spotlight: Navigating Economic Growth and Climate Challenges in a Thriving Industry. mmagroup.co. https://www.linkedin.com/pulse/strawberries-global-spotlight-navigating-economic-growth-mostafa-adel-1u6df/ Accessed 2026-04-07

[2] Robots in Strawberry Fields: How AI Automates Harvesting and Supports Farmers. DFKI GmbH. https://robotik.dfki-bremen.de/en/startpage/news/robots-in-strawberry-fields-how-ai-automates-harvesting-and-supports-farmers# Accessed 2026-04-13

[3] Montoya-Cavero, L.-E., León Torres, R.D., Gómez-Espinosa, A., Cabello, J.A.E.: Vision systems for harvesting robots: Produce detection and localization. Computers and Electronics in Agriculture **192**, 106562 (2022) https://doi.org/10.1016/j.compag.2021.106562

[4] Maró, Z.M., Nagy, J., Molnár, E.M., Mizik, T.: Challenges and potential solutions to employment issues in the agri-food sector of developed countries - a systematic literature review. Sustainable Futures **10**, 100895 (2025) https://doi.org/10.1016/j.sftr.2025.100895

[5] Bagagiolo, G., Matranga, G., Cavallo, E., Pampuro, N.: Greenhouse robots: Ultimate solutions to improve automation in protected cropping systems—a review. Sustainability **14**, 6436 (2022) https://doi.org/10.3390/su14116436

[6] Yurdakul, M., Bastug, Z.S., Gok, A.E., Taşdemir, S.: A Novel Public Dataset for Strawberry (Fragaria x ananassa) Ripeness Detection and Comparative Evaluation of YOLO-Based Models (2026). https://arxiv.org/abs/2602.15656

[7] Chettri, K., Sen, B., Ghosal, P.: Deep learning for precision agriculture: A systematic review of methods, challenges, and future directions. Knowledge and Information Systems **68**(1), 35 (2026) https://doi.org/10.1007/s10115-025-02625-w

[8] Christos Charisis, Dimitrios Argyropoulos: Deep learning-based instance segmentation architectures in agriculture: A review of the scopes and challenges. Smart Agricultural Technology **8**, 100448 (2024) https://doi.org/10.1016/j.atech.2024.100448

[9] Xiao, F., Wang, H., Xu, Y., Zhang, R.: Fruit detection and recognition based on deep learning for automatic harvesting: An overview and review. Agronomy **25**, 3952 (2023) https://doi.org/10.3390/agronomy13061625

[10] Pugazhendi, P., Badgujar, C.M., Sapkota, R., Dhillon, R., Sivaprakash, R., Sheela, J.J., Ganapathy, M.R.: Advances in agricultural fruit detection using you only look once (YOLO) algorithm: A review. Smart Agricultural Technology **13**, 101896 (2026) https://doi.org/10.1016/j.atech.2026.101896

[11] Ma, J., Li, M., Fan, W., Liu, J.: State-of-the-art techniques for fruit maturity detection. Agronomy **14**, 2783 (2024) https://doi.org/10.3390/agronomy14122783

[12] Amjoud, A.B., Amrouch, M.: Object detection using deep learning, CNNs and vision transformers: A review. IEEE Access **11**, 35479–35516 (2023) https://doi.org/10.1109/ACCESS.2023.3266093

[13] Shoaib, M., Shah, B., EI-Sappagh, S., Ali, A., Ullah, A., Alenezi, F., Gechev, T., Hussain, T., Ali, F.: An advanced deep learning models-based plant disease detection: A review of recent research. Frontiers Plant Science **14**, 1158933 (2023) https://doi.org/10.3389/fpls.2023.1158933

[14] Ariza-Sentís, M., Vélez, S., Martínez-Peña, R., Baja, H., Val, J.: Object detection and tracking in precision farming: A systematic review. Computers and Electronics in Agriculture **219**, 108757 (2024) https://doi.org/10.1016/j.compag.2024.108757

[15] Badgujar, C.M., Poulose, A., Gan, H.: Agricultural object detection with you only look once (YOLO) algorithm: A bibliometric and systematic literature

review. Computers and Electronics in Agriculture **223**, 109090 (2024) https://doi.org/10.1016/j.compag.2024.109090

[16] Demilie, W.B.: Plant disease detection and classification techniques: A comparative study of the performances. Journal of Big Data **11**, 5 (2024) https://doi.org/10.1186/s40537-023-00863-9

[17] Remch, Z., Khoulji, S., Kerkeb, M.L.: Object detection techniques for strawberry disease detection: A comprehensive review. In: Journal of Theoretical and Applied Information Technology vol. 19, pp. 7178–7201. https://www.jatit.org/volumes/hundredtwo19.php Accessed 27.03.2026

[18] Upadhyay, A., Chandel, N.S., Singh, K.P., Chakraborty, S.K., Nandede, B.M., Kumar, M., Subeesh, A., Upendar, K., Salem, A., Elbeltagi, A.: Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture. Artificial Intelligence Review **58**(3), 92 https://doi.org/10.1007/s10462-024-11100-x . Accessed 2025-05-19

[19] Hajraoui, N., Azrour, M., Farhaoui, Y., Allaoui, A.E.: A review of AI and deep transfer learning methods for plant disease detection and classification. Salud, Ciencia y Tecnología **6**, 2598 (2026) https://doi.org/10.56294/saludcyt20262598

[20] Muthulakshmi, A., Renjith, P.N.: Comprehensive systematic review on fruit maturity detection technique. In: Second International Conference on Electronics and Sustainable Communication Systems (ICESC), pp. 1234–1240 (2021). https://doi.org/10.1109/ICESC51422.2021.9532772

[21] Mohieddine Jelali: Improved YOLOv8 algorithms for agricultural monitoring and harvesting tasks: A comprehensive review. Preprints (2025) https://doi.org/10.20944/preprints202511.1904.v1

[22] Page, M.J., McKenzie, J.E., Bossuyt, P.M., Boutron, I., Hoffmann, T.C., Mulrow, C.D., Shamseer, L., Tetzlaff, J.M., Akl, E.A., Brennan, S.E., Chou, R., Glanville, J., Grimshaw, J.M., Hróbjartsson, A., Lalu, M.M., Li, T., Loder, E.W., Mayo-Wilson, E., McDonald, S., McGuinness, L.A., Stewart, L.A., Thomas, J., Tricco, A.C., Welch, V.A., Whiting, P., Moher, D.: The prisma 2020 statement: an updated guideline for reporting systematic reviews. BMJ **372** (2021) https://doi.org/10.1136/bmj.n71 https://www.bmj.com/content/372/bmj.n71.full.pdf

[23] He, Q., Wang, S., Wu, C., Xu, J.: An improved lightweight Yolov5 algorithm for strawberry ripeness detection. In: Proceedings - 2023 China Automation Congress, CAC 2023, pp. 3888–3893 (2023). https://doi.org/10.1109/CAC59555.2023.10451290

[24] Pang, F., Chen, X.: MS-YOLOv5: A lightweight algorithm for strawberry

ripeness detection based on deep learning. Systems Science & Control Engineering **11**(1) (2023) https://doi.org/10.1080/21642583.2023.2285292

[25] Li, Y., Xue, J., Zhang, M., Yin, J., Liu, Y., Qiao, X., Zheng, D., Li, Z.: YOLOv5-ASFF: A multistage strawberry detection algorithm based on improved YOLOv5. Agronomy **13**, 1901 (2023) https://doi.org/10.3390/agronomy13071901

[26] Tao, Z., Li, K., Rao, Y., Li, W., Zhu, J.: Strawberry maturity recognition based on improved YOLOv5. Agronomy **14**(3) (2024) https://doi.org/10.3390/agronomy14030460

[27] Tamrakar, N., Karki, S., Kang, M., Deb, N.C., Arulmozhi, E., Kang, D.-y., Kook, J., Kim, H.: Lightweight improved YOLOv5s-CGhostnet for detection of strawberry maturity levels and counting. AgriEngineering **6**(2), 962–978 (2024)

[28] Shen, S., Duan, F., Tian, Z., Han, C.: A novel deep learning method for detecting strawberry fruit. Applied Sciences **14**(10) (2024) https://doi.org/10.3390/app14104213

[29] Huang, L., Li, H., Yang, H., Fu, C., Zhang, Y., Chang, F.: Research on real-time target detection algorithm for strawberry-picking robot based on improved YOLOv5. Journal of Food Measurement and Characterization (2025) https://doi.org/10.1007/s11694-025-03703-x

[30] He, Z., Khanal, S.R., Zhang, X., Karkee, M., Zhang, Q.: Real-time Strawberry Detection Based on Improved YOLOv5s Architecture for Robotic Harvesting in open-field environment (2023)

[31] Du, X., Cheng, H., Ma, Z., Lu, W., Wang, M., Meng, Z., Jiang, C., Hong, F.: DSW-YOLO: A detection method for ground-planted strawberry fruits under different occlusion levels. Computers and Electronics in Agriculture **214** (2023) https://doi.org/10.1016/j.compag.2023.108304

[32] Jin, X., Gu, S., Zhang, R., Yuan, J.: Research on strawberry category detection method based on improved YOLOv7 model. In: ACM International Conference Proceeding Series, pp. 922–927 (2023). https://doi.org/10.1145/3656766.3656919

[33] Tang, Z., Wu, Y., Xu, X.: The study of recognizing ripe strawberries based on the improved YOLOv7-Tiny model. Visual Computer **41**, 3155–3171 (2025) https://doi.org/10.1007/s00371-024-03593-y

[34] Bai, Y., Yu, J., Yang, S., Ning, J.: An improved YOLO algorithm for detecting flowers and fruits on strawberry seedlings. Biosystems Engineering **237**, 1–12 (2024) https://doi.org/10.1016/j.biosystemseng.2023.11.008

[35] Yang, H., Yang, L., Wu, T., Yuan, Y., Li, J., Li, P.: MFD-YOLO: A fast and lightweight model for strawberry growth state detection. Computers and Electronics in Agriculture **234**, 110177 (2025) https://doi.org/10.1016/j.compag.2025.110177

[36] Cheng, J., Wang, Y., Liu, M., Yang, G., Chen, W., Lin, X.: HSK-YOLO: A strawberry maturity detection algorithm based on improved YOLOv8. In: Proceedings of 2024 8th Asian Conference on Artificial Intelligence Technology, ACAIT 2024, pp. 943–947 (2024). https://doi.org/10.1109/ACAIT63902.2024.11022197

[37] Wang, C., Wang, H., Han, Q., Zhang, Z., Kong, D., Zou, X.: Strawberry detection and ripeness classification using YOLOv8+ model and image processing method. Agriculture **14**(5) (2024) https://doi.org/10.3390/agriculture14050751

[38] Chen, Y., Xu, H., Chang, P., Huang, Y., Zhong, F., Jia, Q., Chen, L., Zhong, H., Liu, S.: CES-YOLOv8: Strawberry maturity detection based on the improved YOLOv8. Agronomy **14**(7) (2024) https://doi.org/10.3390/agronomy14071353

[39] Gan, Y., Ren, X., Liu, H., Chen, Y., Lin, P.: Distinguishing difficulty imbalances in strawberry ripeness instances in a complex farmland environment. Applied Sciences **14**(22) (2024) https://doi.org/10.3390/app142210690

[40] Ren, X., Gan, Y., Liu, H., Chen, Y., Lin, P.: Accurately identifying different ripening stages of strawberry fruits in complex agricultural scenarios. International Journal of Fruit Science **25**(1), 62–80 (2025) https://doi.org/10.1080/15538362.2025.2523807

[41] Xu, G., Wang, Z.: A lightweight detection algorithm for strawberry ripeness. IEEE Access **13**, 187614–187626 (2025) https://doi.org/10.1109/ACCESS.2025.3627896

[42] Bai, L., Xia, C., Liu, F., Yang, X., Zhang, T.: Full-dimensional dynamic convolution and progressive learning strategy for strawberry recognition based on YOLOv8. Frontiers in Plant Science **16** (2025) https://doi.org/10.3389/fpls.2025.1541365

[43] Liu, R., Wang, Q., Zhang, H., Wang, L.: An efficient and lightweight YOLOv8s strawberry maturity detection model, pp. 14–22 (2025). https://doi.org/10.1145/3718441.3718444

[44] Yue, Y., Xu, S., Wu, H.: A strawberry ripeness detection method based on improved YOLOv8. Applied Sciences **15**(11) (2025) https://doi.org/10.3390/app15116324

[45] Yu, H., Qian, C., Chen, Z., Chen, J., Zhao, Y.: Ripe-Detection: A lightweight method for strawberry ripeness detection. Agronomy **15**(7) (2025) https://doi.org/10.3390/agronomy15071645

[46] Ogundele, O.M., Tamrakar, N., Kook, J., Kim, S.-m., Choi, J.-i., Karki, S., Akpenpuun, T.D., Kim, H.: Real-time strawberry ripeness classification and counting: An optimized YOLOv8s framework with class-aware multi-object tracking. Agriculture **15**(18) (2025) https://doi.org/10.3390/agriculture15181906

[47] Zhao, S., Fang, C., Hua, T., Jiang, Y.: Detecting the maturity of red strawberries using improved YOLOv8s model. Agriculture **15**(21) (2025) https://doi.org/10.3390/agriculture15212263

[48] Tao, Z., Li, K., Rao, Y., Li, W., Zhu, J.: Lightweight model for simultaneous detection of strawberries maturity and picking points based on improved yolov8. Journal of Food Measurement and Characterization (2026) https://doi.org/10.1007/s11694-026-04048-9

[49] Ma, Z., Dong, N., Gu, J., Cheng, H., Meng, Z., Du, X.: STRAW-YOLO: A detection method for strawberry fruits targets and key points. Computers and Electronics in Agriculture **230**, 109853 (2025) https://doi.org/10.1016/j.compag.2024.109853

[50] He, L., Wu, D., Zheng, X., Xu, F., Lin, S., Wang, S., Ni, F., Zheng, F.: RLK-YOLOv8: Multi-stage detection of strawberry fruits throughout the full growth cycle in greenhouses based on large kernel convolutions and improved YOLOv8. Frontiers in Plant Science **16**, 1552553 (2025) https://doi.org/10.3389/fpls.2025.1552553

[51] Gao, S., Cui, G., Wang, Q.: WCS-YOLOv8s: an improved YOLOv8s model for target identification and localization throughout the strawberry growth process. Frontiers in Plant Science **16** (2025) https://doi.org/10.3389/fpls.2025.1579335

[52] Dai, S., Bai, T., Zhao, Y.: Keypoint detection and 3D localization method for ridge-cultivated strawberry harvesting robots. Agriculture **15**(4) (2025) https://doi.org/10.3390/agriculture15040372

[53] Xie, H., Zhang, Z., Zhang, K., Yang, L., Zhang, D., Yu, Y.: Research on the visual location method for strawberry picking points under complex conditions based on composite models. Journal of the Science of Food and Agriculture **104**(14), 8566–8579 (2024) https://doi.org/10.1002/jsfa.13684

[54] Luo, Q., Wu, C., Li, W.: A small target strawberry recognition method based on improved YOLOv8n model. IEEE Access **12**, 14987–14995 (2024) https://doi.org/10.1109/ACCESS.2024.3356869

[55] Teng, H., Sun, F., Wu, H., Lv, D., Lv, Q., Feng, F., Yang, S., Li, X.: DS-YOLO: A lightweight strawberry fruit detection algorithm. Agronomy **15**(9) (2025) https://doi.org/10.3390/agronomy15092226

[56] Zhang, X., Zhang, G., Wang, J., Yang, J., Ge, Q., Zhao, R., Wang, Y.: Efficient instance segmentation for strawberry in greenhouses using yolov8n-mcp on edge devices. Information Processing in Agriculture **12**(4), 539–549 (2025) https://doi.org/10.1016/j.inpa.2025.07.001

[57] He, Z., Liu, Z., Zhou, Z., Karkee, M., Zhang, Q.: Improving picking efficiency under occlusion: Design, development, and field evaluation of an innovative robotic strawberry harvester. Computers and Electronics in Agriculture **237**, 110684 (2025) https://doi.org/10.1016/j.compag.2025.110684

[58] Liu, B., Chen, C., Li, J., Zhang, Q., Meng, Y.: Research on three-dimensional positioning method for automatic strawberry fruit picking based on vision–IMU fusion. Agriculture **16**(8) (2026) https://doi.org/10.3390/agriculture16080893

[59] Mo, Y., Bai, S., Chen, W.: ASHM-YOLOv9: A detection model for strawberry in greenhouses at multiple stages. Applied Sciences **15**(15) (2025) https://doi.org/10.3390/app15158244

[60] Ye, R., Shao, G., Zhang, H., Li, T., Gao, Q.: Strawberry maturity detection algorithm based on improved YOLOv9. In: Proceedings - 2024 International Conference on Advances in Electrical Engineering and Computer Applications, AEECA 2024, pp. 759–766 (2024). https://doi.org/10.1109/AEECA62331.2024.00133

[61] Ye, R., Shao, G., Gao, Q., Zhang, H., Li, T.: CR-YOLOv9: Improved YOLOv9 multi-stage strawberry fruit maturity detection application integrated with CRNET. Foods **13**(16) (2024) https://doi.org/10.3390/foods13162571

[62] Li, Z., Wang, J., Gao, G., Lei, Y., Zhao, C., Wang, Y., Bai, H., Liu, Y., Guo, X., Li, Q.: SGSNet: a lightweight deep learning model for strawberry growth stage detection. Frontiers in Plant Science **Volume 15 - 2024** (2024) https://doi.org/10.3389/fpls.2024.1491706

[63] Ma, H., Zhao, Q., Zhang, R., Hao, C., Dong, W., Zhang, X., Li, F., Xue, X., Sun, G.: YOLOv11-GSF: An optimized deep learning model for strawberry ripeness detection in agriculture. Frontiers in Plant Science **16** (2025) https://doi.org/10.3389/fpls.2025.1584669

[64] Fu, H., Li, X., Li, Z., Zhu, L., Feng, Y.: LBS-YOLO: A lightweight model for strawberry ripeness detection. Frontiers in Plant Science **16** (2025) https://doi.org/10.3389/fpls.2025.1715263

[65] Liu, J., Guo, J., Zhang, S.: YOLOv11-HRS: An improved model for strawberry ripeness detection. Agronomy **15**(5) (2025) https://doi.org/10.3390/agronomy15051026

[66] Li, Z., Hu, X., Zhao, X., Ye, H., Chen, F., Chen, X., Li, X.: Beyond obstacles:

Feather-light YOLO11-LES for real-time ripeness detection of occluded strawberries in greenhouses. Journal of Real-Time Image Processing **22**(5) (2025) https://doi.org/10.1007/s11554-025-01748-1

[67] Zhen, J., Wang, Y., Liu, X., Zhou, L., Yang, S., Wang, D.: Detection of strawberry bloom phenology based on YOLO-RCMC and flower opening scale. Smart Agricultural Technology **12**, 101275 (2025) https://doi.org/10.1016/j.atech.2025.101275

[68] Xu, D., Li, Z., Zhang, C., Ouyang, G., Chen, Y., Xu, W., Qu, R., Lin, F., Tan, S.: Real-time detection of multi-growth-stage strawberry in complex orchard environments via smartphone. Smart Agricultural Technology **12**, 101468 (2025) https://doi.org/10.1016/j.atech.2025.101468

[69] Gan, Y., Ren, X., Liu, H., Chen, Y., Lin, P.: Real-time lightweight strawberry ripeness detection framework based on YOLO11 deployed on edge computing platform. Journal of Food Measurement and Characterization **19**(11), 8469–8491 (2025) https://doi.org/10.1007/s11694-025-03559-1

[70] Hao, H., Xi, J., Dai, J., Wang, G., Liu, D., Zhu, L.: DHN-YOLO: A joint detection algorithm for strawberries at different maturity stages and key harvesting points. Plants **14**(22) (2025) https://doi.org/10.3390/plants14223439

[71] Li, K., Wei, X., Wang, Q., Zhang, W.: Research on strawberry visual recognition and 3D localization based on lightweight RAFS-YOLO and RGB-D camera. Agriculture **15**(21) (2025) https://doi.org/10.3390/agriculture15212212

[72] Zhao, Q., Hao, C., Cui, J., Zan, J., Han, X., Chen, Q., Zhang, X., Li, F.: B2G-YOLO11-S: An efficient intelligent grading model for strawberry maturity with integrated causal analysis. Smart Agricultural Technology **12** (2025) https://doi.org/10.1016/j.atech.2025.101607

[73] Jiang, L., Wang, Y., Yan, H., Yin, Y., Wu, C.: Strawberry fruit deformity detection and symmetry quantification using deep learning and geometric feature analysis. Horticulturae **11**(6) (2025) https://doi.org/10.3390/horticulturae11060652

[74] Zhao, Q., Zhang, R., Cui, J., Han, X., Zhang, L., Chen, Q., Li, F., Zhang, X.: YOLO11-SCC: Causal-guided semi-supervised deep learning for strawberry maturity detection. Scientia Horticulturae **360**, 114794 (2026) https://doi.org/10.1016/j.scienta.2026.114794

[75] Leng, T., Zhang, Y., Fan, Y., Gao, H., Zhou, B., Mu, J.: A lightweight shape-aware yolo network for field strawberry maturity detection under complex orchard conditions. Agronomy **16**(11) (2026) https://doi.org/10.3390/agronomy16111074

[76] Zhou, X., Zhang, Y., Jiang, X., Riaz, K., Rosenbaum, P., Lefsrud, M., Sun, S.: Advancing tracking-by-detection with MultiMap: Towards occlusion-resilient online multiclass strawberry counting. Expert Systems with Applications **255**, 124587 (2024) https://doi.org/10.1016/j.eswa.2024.124587

[77] Yang, S., Wang, W., Gao, S., Deng, Z.: Strawberry ripeness detection based on YOLOv8 algorithm fused with LW-Swin Transformer. Computers and Electronics in Agriculture **215** (2023) https://doi.org/10.1016/j.compag.2023.108360

[78] Du, X., Huang, W., Min, H.: SDSLP-YOLO: An end-to-end model for simultaneously detecting strawberry and locating picking point. In: 2024 China Automation Congress (CAC), pp. 6098–6103 (2024). https://doi.org/10.1109/CAC63892.2024.10865495

[79] Xia, C.: Rapid strawberry ripeness detection and 3D localization of picking point based on improved YOLO V8-Pose with RGB-camera. Journal of Electrical Systems **20-3s**, 2171–2181 (2024) https://doi.org/10.52783/jes.1840

[80] Zhao, C., Lei, Y., Wang, J., Li, Z., Li, J., Wang, Y., Bai, H.: DynaFog-BerryNet: A unified model for image dehazing and object detection toward accurate strawberry maturity monitoring. Smart Agricultural Technology **12** (2025) https://doi.org/10.1016/j.atech.2025.101335

[81] Wu, J., Zhang, J., Tan, L., Wu, Y., Gao, K.: Lightweight transformer and faster convolution for efficient strawberry detection. Applied Sciences **16**, 293 (2026) https://doi.org/10.3390/app16010293

[82] Mi, Z., Yan, W.: Strawberry ripeness detection using deep learning models. Big Data and Cognitive Computing **8**(8) (2024) https://doi.org/10.3390/bdcc8080092

[83] StrawDI – The Strawberry Digital Images Data Set. https://strawdi.github.io/ Accessed 2026-03-09

[84] El-Bendary, N., Elhariri, E.: Strawberry-DS. https://data.mendeley.com/datasets/z6dtfdpzz8/1 Accessed 2026-03-09

[85] Liu, H., Gan, Y., Ren, X., Xu, J., Lin, P., Chen, Y.: Vision-based quantitative assessment of strawberry ripeness using a detail-aware perception transformer in complex agricultural environments. Journal of Food Measurement and Characterization (2026) https://doi.org/10.1007/s11694-026-04078-3

[86] Liu, H., Wang, X., Zhao, F., Yu, F., Lin, P., Gan, Y., Ren, X., Chen, Y., Tu, J.: Upgrading swin-B transformer-based model for accurately identifying ripe strawberries by coupling task-aligned one-stage object detection mechanism. Computers and Electronics in Agriculture **218** (2024) https://doi.org/10.1016/j.compag.2024.108674

[87] Final Strawberry Ripeness Dataset. https://www.kaggle.com/datasets/aishtrips/final-strawberry-ripeness-dataset Accessed 2026-03-03

[88] New Strawberries. https://aistudio.baidu.com/datasetdetail/147119 Accessed 2026-03-09

[89] Chen, R., Xie, Z., Lin, C.: YOLO-ODM based rapid detection of strawberry ripeness in greenhouse. Journal of Huazhong Agricultural University **42**(4), 262–269 (2023) https://doi.org/10.13300/j.cnki.hnlkxb.2023.04.030

[90] Lu, Y., Young, S.: A survey of public datasets for computer vision tasks in precision agriculture. Computers and Electronics in Agriculture **178**, 105760 (2020) https://doi.org/10.1016/j.compag.2020.105760

[91] Li, Z., Wang, R., Ding, R.: A review of crop attribute monitoring technologies for general agricultural scenarios. AgriEngineering **7**(11), 365 (2025) https://doi.org/10.3390/agriengineering7110365

[92] Ngugi, H.N., Ezugwu, A.E., Akinyelu, A.A., Abualigah, L.: Revolutionizing crop disease detection with computational deep learning: A comprehensive review. Environmental monitoring and assessment **196**(3), 302 (2024) https://doi.org/10.1007/s10661-024-12454-z

[93] Sunil, C.K., Jaidhar, C.D., Patil, N.: Systematic study on deep learning-based plant disease detection or classification. Artificial Intelligence Review **56**(12), 14955–15052 (2023) https://doi.org/10.1007/s10462-023-10517-0

[94] Zheng, C., Abd-Elrahman, A., Whitaker, V.: Remote sensing and machine learning in crop phenotyping and management, with an emphasis on applications in strawberry farming. Remote Sensing **13**(3), 531 (2021) https://doi.org/10.3390/rs13030531

[95] Pérez-Borrero, I., Marín-Santos, D., Gegúndez-Arias, M.E., Cortés-Ancos, E.: A fast and accurate deep learning method for strawberry instance segmentation. Computers and Electronics in Agriculture **178**, 105736 (2020) https://doi.org/10.1016/j.compag.2020.105736

[96] Ajayi, O.G., Ashi, J., Guda, B.: Performance evaluation of YOLO v5 model for automatic crop and weed classification on uav images. Smart Agricultural Technology **5**, 100231 (2023) https://doi.org/10.1016/j.atech.2023.100231

[97] Wu, Q., Zhang, S., Wan, J., Wang, X., Guo, Z., Liu, M., Weng, S., Jia, W.: Synchronous detection of strawberry fruit and stem in natural scene using scale edge fusion network and semi-supervised learning. Engineering Applications of Artificial Intelligence **163** (2026) https://doi.org/10.1016/j.engappai.2025.112916

[98] Pagire, V., Chavali, M., Kale, A.: A comprehensive review of object detection with traditional and deep learning methods. Signal Processing **237**, 110075 (2025) https://doi.org/10.1016/j.sigpro.2025.110075

[99] Goodfellow, I., Courville, A., Bengio, Y.: Deep Learning. Adaptive computation and machine learning. The MIT Press, Cambridge, Massachusetts (2016). http://www.deeplearningbook.org

[100] Szeliski, R.: Computer Vision: Algorithms and Applications. Texts in Computer Science. Springer. https://doi.org/10.1007/978-3-030-34372-9 . https://link.springer.com/10.1007/978-3-030-34372-9 Accessed 2025-06-28

[101] Padilla, R., Netto, S.L., Silva, E.A.B.: A survey on performance metrics for object-detection algorithms. In: 2020 International Workshop on Systems, Signal Processing and Their Applications (IWSSIP), pp. 237–242. IEEE, Niterói, Brazil (2020). https://doi.org/10.1109/IWSSIP48289.2020.9145130 . https://ieeexplore.ieee.org/document/9145130

[102] Yuan, W.: Odei: Object detector efficiency index. AI **6**(7) (2025) https://doi.org/10.3390/ai6070141

[103] Molchanov, P., Tyree, S., Karras, T., Aila, T., Kautz, J.: Pruning convolutional neural networks for resource efficient transfer learning. ArXiv **abs/1611.06440** (2016)

[104] Sapkota, R., Ahmed, D., Karkee, M.: Comparing YOLOv8 and Mask R-CNN for instance segmentation in complex orchard environments. Artificial Intelligence in Agriculture **13**, 84–99 (2024) https://doi.org/10.1016/j.aiia.2024.07.001

[105] Fabian Gerz, Mohieddine Jelali: Towards reliable data augmentation in machine learning: Practices to prevent data leakage. Smart Agricultural Technology (2025) https://doi.org/10.2139/ssrn.5636100

[106] Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. Journal of Big Data **6**(1) (2019) https://doi.org/10.1186/s40537-019-0197-0

[107] Zhang, Y., Zhang, L., Yu, H., Guo, Z., Zhang, R., Zhou, X.: Research on the strawberry recognition algorithm based on deep learning. Applied Sciences **13**(20) (2023) https://doi.org/10.3390/app132011298

[108] Kaufman, S., Rosset, S., Perlich, C., Stitelman, O.: Leakage in data mining. ACM Transactions on Knowledge Discovery from Data **6**(4), 1–21 (2012) https://doi.org/10.1145/2382577.2382579

[109] Apicella, A., Isgrò, F., Prevete, R.: Don't push the button! exploring data leakage risks in machine learning and transfer learning. Artificial Intelligence Review

58(11) (2025) https://doi.org/10.1007/s10462-025-11326-3

[110] Perez, L., Wang, J.: The Effectiveness of Data Augmentation in Image Classification using Deep Learning. https://arxiv.org/pdf/1712.04621

[111] T. Kumar, R. Brennan, A. Mileo, M. Bendechache: Image data augmentation approaches: A comprehensive survey and future directions. IEEE Access **12**, 187536–187571 (2024) https://doi.org/10.1109/ACCESS.2024.3470122

[112] Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: RandAugment: Practical automated data augmentation with a reduced search space. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20. Curran Associates Inc, Red Hook, NY, USA (2020)

[113] Joshi, K., Hooda, S., Sharma, A., Sonah, H., Deshmukh, R., Tuteja, N., Gill, S.S., Gill, R.: Precision diagnosis of tomato diseases for sustainable agriculture through deep learning approach with hybrid data augmentation. Current Plant Biology **41**, 100437 (2025) https://doi.org/10.1016/j.cpb.2025.100437

[114] Yin, J., Zhu, J., Chen, G., Jiang, L., Zhan, H., Deng, H., Long, Y., Lan, Y., Wu, B., Xu, H.: An intelligent field monitoring system based on enhanced YOLO-RMD architecture for real-time rice pest detection and management. Agriculture **15**(8), 798 (2025) https://doi.org/10.3390/agriculture15080798

[115] Wong, S.C., Gatt, A., Stamatescu, V., McDonnell, M.D.: Understanding data augmentation for classification: When to warp? https://arxiv.org/pdf/1609.08764

[116] Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., Rieck, K.: Dos and don'ts of machine learning in computer security. In: 31st USENIX Security Symposium (USENIX Security 22), pp. 3971–3988. USENIX Association, Boston, MA (2022). https://www.usenix.org/conference/usenixsecurity22/presentation/arp

[117] Ranjan Sapkota, Marco Flores-Calero, Rizwan Qureshi, Chetan Badgujar, Upesh Nepal, Alwin Poulose, Peter Zeno, Uday Bhanu Prakash Vaddevolu, Sheheryar Khan, Maged Shoman, Hong Yan, Manoj Karkee: YOLO advances to its genesis: A decadal and comprehensive review of the You Only Look Once (YOLO) series. Artificial Intelligence Review **58**, 274 (2025) https://doi.org/10.1007/s10462-025-11253-3

[118] Rosada, R., Hussein, Z.M., Novamizanti, L.: Evaluating yolo variants for real-time multi-object detection of strawberry quality and ripeness, pp. 483–490 (2025). https://doi.org/10.1109/IAICT65714.2025.11100674

[119] Singh, R., Gadade, A.M., Hussain, I.: Robust real-time strawberry maturity detection using UAV-mounted deep learning for precision agriculture. BMC

Plant Biology **25**(1) (2025) https://doi.org/10.1186/s12870-025-07246-7

[120] Aktouf, L., Shivanna, Y., Dhimish, M.: High-precision defect detection in solar cells using YOLOv10 deep learning model. Solar **4**(4), 639–659 (2024) https://doi.org/10.3390/solar4040030

[121] Sulake, N.R.: YOLOv11 Demystified: A Practical Guide to High-Performance Object Detection. https://arxiv.org/abs/2604.03349v1

[122] Yang, G., Wu, Y., Weng, D., Zeng, L.: Strawberry ripeness detection in complex environment based on improved RT-DETR. Agronomy Journal **117**(5) (2025) https://doi.org/10.1002/agj2.70162

[123] Zhang, Y., Fang, Z., Liao, Y., Lu, J., Zeng, S., Du, J., Chen, B.: Strawberry ripeness detection based on CRB model of YOLOv10. In: Proceedings of SPIE - The International Society for Optical Engineering, vol. 13985 (2025). https://doi.org/10.1117/12.3083520

[124] Du, X., Huang, W., Min, H.: SDSLP-YOLO: An end-to-end model for simultaneously detecting strawberry and locating picking point, pp. 6098–6103 (2024). https://doi.org/10.1109/CAC63892.2024.10865495

[125] Li, Y., Wang, W., Guo, X., Wang, X., Liu, Y., Wang, D.: Recognition and positioning of strawberries based on improved YOLOv7 and RGB-D sensing. Agriculture **14**(4) (2024) https://doi.org/10.3390/agriculture14040624

[126] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (2021)

[127] Saber Mehdipour, Seyed Abolghasem Mirroshandel, Seyed Amirhossein Tabatabaei: Vision transformers in precision agriculture: A comprehensive survey. Intelligent Systems with Applications **29**, 200617 (2026) https://doi.org/10.1016/j.iswa.2025.200617

[128] Gerz, F., Schneider, M.G., Jelali, M.: Integration of vision transformer networks in YOLOv8 for object detection: Comparative study on plant disease detection. IEEE Access **14**, 27303–27338 (2026) https://doi.org/10.1109/ACCESS.2026.3665969

[129] Friedman, D., Dieng, A.B.: The vendi score: A diversity evaluation metric for machine learning. Transactions on Machine Learning Research (2023)

[130] Nisbet, R., Elder, J., Miner, G.: Chapter 20 - top 10 data mining mistakes. In: Nisbet, R., Elder, J., Miner, G. (eds.) Handbook of Statistical Analysis and Data Mining Applications, pp. 733–754. Academic Press, Boston (2009). https:

//doi.org/10.1016/B978-0-12-374765-5.00020-6

[131] Tribiakto, H., Sunyoto, A., Pramono, E.: Enhancing deep learning-based classification of cassava leaf diseases using clahe and smote. SISTEMASI **14**(6), 2698 (2025) https://doi.org/10.32520/stmsi.v14i6.5530

[132] S. M. H. Rizvi, A. Naseer, S. U. Rehman, S. Akram, V. Gruhn: Revolutionizing agriculture: Machine and deep learning solutions for enhanced crop quality and weed control. IEEE Access **12**, 11865–11878 (2024) https://doi.org/10.1109/ACCESS.2024.3355017

[133] Wang, D., Wang, X., Chen, Y., Wu, Y., Zhang, X.: Strawberry ripeness classification method in facility environment based on red color ratio of fruit rind. Computers and Electronics in Agriculture **214** (2023) https://doi.org/10.1016/j.compag.2023.108313

[134] Jiajun Li, Zifeng Zhu, Hongxin Liu, Yurong Su, Limiao Deng: Strawberry R-CNN: Recognition and counting model of strawberry based on improved faster R-CNN. Ecological Informatics **77**, 102210 (2023) https://doi.org/10.1016/j.ecoinf.2023.102210

[135] Duan, S., Tian, G., Wang, Z., Liu, S., Feng, C.: A semantic robotic grasping framework based on multi-task learning in stacking scenes. Engineering Applications of Artificial Intelligence **121**, 106059 (2023) https://doi.org/10.1016/j.engappai.2023.106059

[136] Yokota, Y., Suzuki, K., Kanazawa, Y., Takebayashi, T.: A multi-task learning framework for grasping-position detection and few-shot classification. In: 2020 IEEE/SICE International Symposium on System Integration (SII), pp. 1033–1039 (2020). https://doi.org/10.1109/SII46433.2020.9025946

[137] Sireesha, N.V., Rekha, G., Almenweer, R.A.: Unassailable citrus disease classification via multi-stage deep ensemble learning with vision transformers. Scientific Reports **15**(1), 40573 (2025) https://doi.org/10.1038/s41598-025-24416-7