Open Peer Review on Qeios

RESEARCH ARTICLE

Comprehensive Performance Evaluation of YOLO11, YOLOv10, YOLOv9 and YOLOv8 on Detecting and Counting Fruitlet in Complex Orchard Environments

Ranjan Sapkota[1], Zhichao Meng[2], Martin Churuvija[1], Xiaoqiang Du[2], Zenghong Ma[2], Manoj Karkee[1]

1 Center for Precision & Automated Agricultural Systems, Washington State University, United States
2 School of Mechanical Engineering, Zhejiang Sci-Tech University, China

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.

Abstract

Object detection, specifically fruitlet detection, is a crucial image processing technique in agricultural automation, enabling the accurate identification of fruitlets on orchard trees within images. It is vital for early fruit load management and overall crop management, facilitating the effective deployment of automation and robotics to optimize orchard productivity and resource use. This study systematically performed an extensive evaluation of the performances of all configurations of YOLOv8, YOLOv9, YOLOv10, and YOLO11 object detection algorithms in terms of precision, recall, mean Average Precision at 50% Intersection over Union (mAP@50), and computational speeds including pre-processing, inference, and post-processing times immature green apple (or fruitlet) detection in commercial orchards. Additionally, this research performed and validated in-field counting of fruitlets using an iPhone and machine vision sensors in 4 different apple varieties (Scifresh, Scilate, Honeycrisp & Cosmic crisp). This investigation of total 22 different configurations of YOLOv8, YOLOv9, YOLOv10 and YOLO11 (5 for YOLOv8, 6 for YOLOv9, 6 for YOLOv10, and 5 for YOLO11) revealed that YOLOv9 gelan-base and YOLO11s outperforms all other configurations of YOLOv10, YOLOv9 and YOLOv8 in terms of mAP@50 with a score of 0.935 and 0.933 respectively. In terms of precision, specifically, YOLOv9 Gelan-e achieved the highest mAP@50 of 0.935, outperforming YOLOv11s's 0.0.933, YOLOv10s’s 0.924, and YOLOv8s's 0.924. In terms of recall, YOLOv9 gelan-base achieved highest value among YOLOv9 configurations (0.899), and YOLO11m performed the best among the YOLO11 configurations (0.897). In comparison for inference speeds, YOLO11n demonstrated fastest inference speeds of only 2.4 ms, while the fastest inference speed across YOLOv10, YOLOv9 and YOLOv8 were 5.5, 11.5 and 4.1 ms for YOLOv10n, YOLOv9 gelan-s and YOLOv8n respectively.

1. Introduction

Object detection in commercial orchards is the foundation to developing agricultural automation and robotics solutions for labor intensive tasks such as harvesting, thinning, and pruning[1][2][3]. One such labor-intensive operations in apple orchards is thinning green fruit in their early growth stage (fruitlets), which is crucial due to its role in enhancing crop yield and quality. Automating fruitlet thinning process is essential for minimizing the dependence of rapidly depleting farm labor, which requires a robust machine vision system for fruitlet detection and localization in orchard environments[4].

Most of the tree fruit crops often set a greater number of fruit per tree than the desired number which causes fruit-to-fruit competition for water, sunlight, and nutrients, resulting inadequate exposure of the fruits to the sun, less space to grow, and overall reduced fruit quality[4]. Additionally, too many fruits can result in reduced cold hardiness, breakage of tree limbs, and exhaustion of tree reserves[5]. Fruitlet thinning in the commercial production of tree fruit crops such as apples, kiwifruit, pears, peaches and plums has been practiced for thousands of years[5] to address these challenges and ensure optimal fruit size and quality[6][7].

Apples is the third most consumed fruit in the United States (U.S.). U.S. also produces an average of 4.6 million tons of apples yearly, making the country the world's second-largest contributor in apple production[8][9]. Around 382 thousand acres of land in the U.S. is used for farming apples commercially, which contributes to exporting 42 million bushels of apples worldwide with an estimated downstream value of $21 billion dollars each year[10]. In the U.S., approximately 1.5 million hired workers are employed in agriculture annually, with about three-fourths or 1.1 million working in crop-production activities[11]. One of the most labor-intensive operations in apple production is fruitlet thinning, and therefore developing robotic fruitlet thinning solutions is essential for sustainable apple production in U.S and around the world.[11]

The availability of farm labor has continued to decline over the past decade, a situation worsened by the global pandemic, which led to an estimated loss of $309 million in agricultural production from March 2020 to March 2021[12][13]. On the other hand, global urbanization has been transforming rural areas, causing 68% of the population to reside in urban environments by 2050 (United Nations). which would further increase the labor shortage in agriculture. On top of that, labor costs on specialty crop farms in the United States are 3 times higher than the average cost of labor in all U.S. farms (USDA, 2018), which further emphasizes the need for automating labor-intensive operations in apple and other specialty crop fields.

In commercial apple orchards, the demand for labor-intensive fruitlet thinning peaks during the summer months. Figure 1a demonstrates the overcrowding of fruitlet in tree branches as each flower cluster leads to several fruits requiring fruitlet thinning to optimize the number of fruit in each branch based the branch diameter. sunlight, space, and water. Figure 1b shows farm workers manually thinning apple clusters, a process that is both time-consuming and laborious, requiring a significant number of seasonal workers.

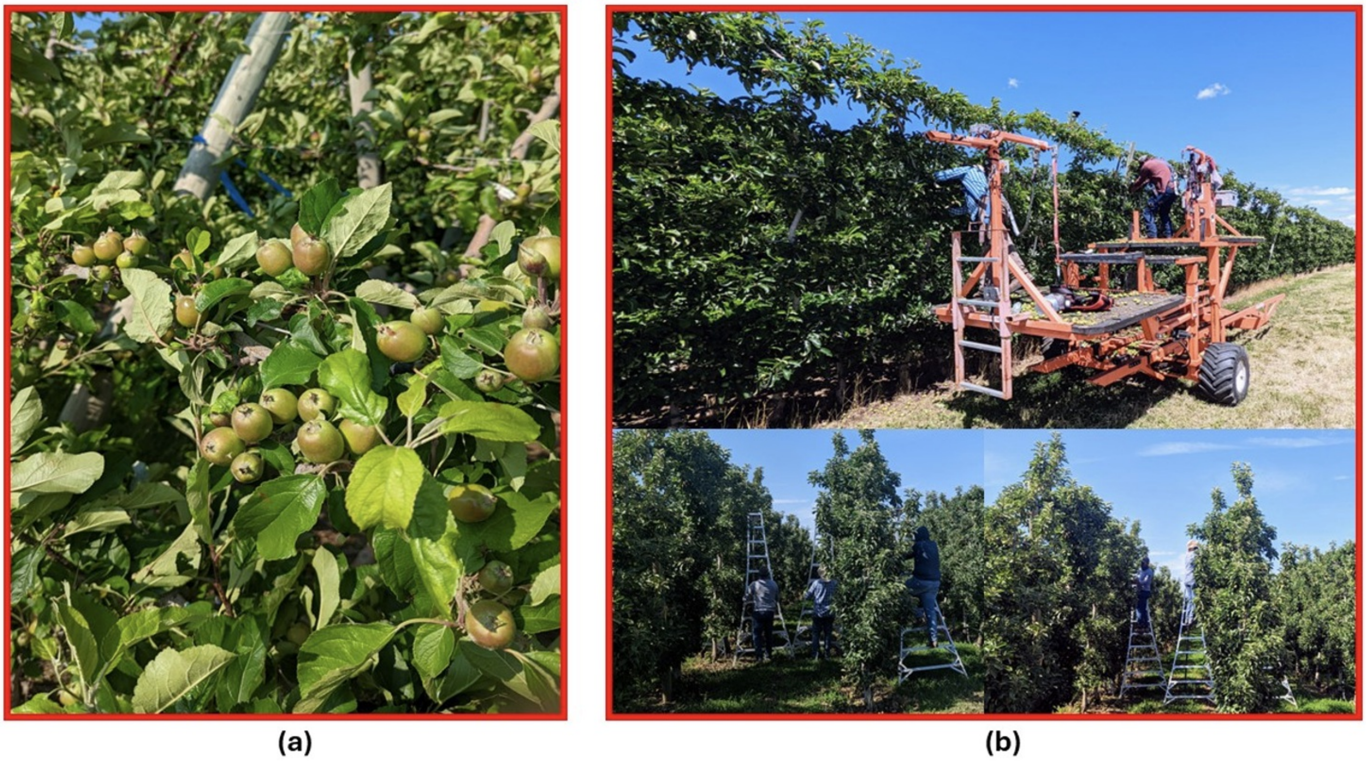


Figure 1. Fruitlet thinning in commercial orchards: (a) High-density clusters of apple fruitlets on a Scilate apple tree during the peak thinning period in June 2022 (A commercial orchard in Prosser, WA), illustrating the typical overcropping seen in commercial orchards; (b) Top: Laborers utilizing an height-adjustable platform for efficiently thinning fruitlet in various parts of tree canopies; b) Bottom: A worker in a Scifresh apple orchard manually thinning excess fruitlets using an aluminum ladder, a common practice that highlights the labor-intensive nature of this crucial agricultural task.

As the U.S. agricultural workforce continues to decline, maintaining competitiveness in the global market necessitates the adoption of labor-saving technologies. Agricultural robots, which can replicate human tasks such as fruitlet thinning during the early growing season, offer a promising solution to labor-intensive operations[14]. In addition to reducing dependance on manual labor, automated and robotic systems for fruit thinning can also contribute to enhancing worker health and safety. Occupational health studies highlight the significant risks associated with manual labor in orchards. For instance, May et al.[15] reveals that farmworkers suffer from a high incidence of musculoskeletal injuries, skin diseases, and other health issues, with the agricultural sector experiencing a fatality rate seven times the national average. Further supporting this need, the study by Earle-Richardson et al.[16] documents the detrimental effects of prolonged physical labor on migrant orchard workers, demonstrating significant decreases in muscle strength within a single workday.[17]

The first crucial step in automating the fruit thinning process is developing a robust vision system capable of detecting green apple fruitlets during the early thinning stage in commercial orchards. Green apple detection is fundamentally an object detection challenge, a domain extensively studied within computer vision. Apple detection systems in orchard environments for possible automation have been reported since the late 90s[17]. In the past, many studies have reported apple detection models for supporting robotic harvesting using traditional image processing techniques such as image color segmentation in RGB, HIS (hue, saturation, and intensity), and/or HSL (hue, saturation, luminance) color spaces[18][19][20][21]. However, the accuracy of these traditional methods has been relatively low, and the approach is limited to a specific environmental condition such as uniform lighting and background conditions created artificially in orchards. It is challenging to implement these image processing algorithms to detect apples in a natural environment with high level of occlusion, fluctuating illumination, and complex backgrounds[22].

During the last few years, machine learning (ML) has become widely adopted for object detection in agriculture, providing more robust detection capabilities with reduced level of image pre-processing and feature extraction[23]. Some of the recent ML models used in detecting apples have been Simple Linear Iterative Clustering (SLIC), Support Vector Machine (SVM) and Random Forest (RF) to detect apples[24][25]. Additionally, deep learning (DL)-based models have become one of the most effective methods for performing object detection in agriculture as it provides end-to-end processing capability without any manual feature extraction. To perform apple detection in orchards, researchers have implemented DL models such as Faster R-CNN models based on AlexNet and based on ResNet101[26]. More recently, Kang et al.[27] have purposed a deep-learning-based detector called ‘LedNet’, which is able to perform real-time and accurate apple detection in orchards. The LedNet architecture utilizes the 3-levels feature pyramid network (FPN) and Atrous Spatial Pyramid Pooling (ASPP), a semantic segmentation module used in the feature processing block of the DL module.

Although researchers have been successful in detecting mature apples for robotic harvesting, there are only a few reported studies on the detection of apples during their early growing season for robotic thinning applications. During the early growing season of apples in the natural environment, the color of apples and light reflection is similar to that of leaves, which poses a great challenge in accurately detecting apple fruitlet. Additionally, factors such as occlusion and overlap of fruits with leaves and branches, and uncertain illumination make it difficult to detect fruitlet effectively[28][29].

Xia et al.[30] used improved Hough transform algorithm and SVM to detect green apples in a natural environment and reported an F1 score of 90.3%. The author used an iterative threshold segmentation (ITS) algorithm to detect regions of interest (ROI) that contained potential apple fruitlet pixels. However, this technique was limited to detecting only non-overlapping apples. Recently, to detect young apples, Tian et. al[31] implemented a YOLOv3 model called ‘Densenet’ and reported an F1 score of 83.2%. Additionally, to detect apples, Huang et al.[32] recently used an improved YOLOv3 model based on the CSPDarknet53 DL network. The study reported an inference speed of 8.6 ms in detecting immature apples. However, F1 and mAP achieved by the proposed model were only 0.65 and 0.67. Most recently, Wang at al.[33] presented a YOLOv5s-based deep learning technique to detect apple fruitlet before thinning activity in the natural environment. The author of this study implemented transfer learning through a channel pruned YOLOv5s object detection model and fine-tuned the model to achieve a higher detection rate. However, the model size implemented in this study was very small (1.4MB) and the author reported a false detection of 4.2%.

Recent advancements in deep learning models for object detection have significantly enhanced fruitlet detection performance in field environments. The You Only Look Once (YOLO) framework, modified for specific agricultural needs, has shown promising results. For instance,[34] reported that a tailored YOLO model improved the detection of green citrus in complex orchard environments with a substantially better accuracy and speed compared to YOLO v4[35]. Similarly, a lightweight version of YOLOv8 was developed for detecting pomegranate fruitlets, achieving high precision with reduced model complexity, making it suitable for mobile devices[36]. Additionally, a channel-pruned YOLO V5s model effectively detected apple fruitlets under varying conditions, outperforming several methods while maintaining a compact size for potential use in mobile fruit thinning terminals[37].

Similarly, an adaptation of the YOLOX-m model, optimized for robust feature extraction and enhanced by a feature fusion pyramid network with an Atrous Spatial Pyramid Pooling (ASPP) module, has shown improvements in detecting fruitlets. The ASPP module-based network achieved an average precision of 64.3% for apples and 74.7% for persimmons, outperforming several common detection models and providing a solid reference for diverse fruit and vegetable detection tasks[34]. Similarly, the YOLO-P model, a modified version of YOLOv5 designed specifically for pear detection in orchards, incorporated advanced architectural changes such as shuffle blocks and a convolutional block attention module (CBAN). These modifications improved feature extraction capabilities, allowing the model to perform well under challenging environmental conditions including complex backgrounds and varied lighting. This model not only supports rapid and precise pear detection but also offers insights for enhancing fruit detection systems in similar unstructured environments[38].

Moreover, a YOLO v4-based method specifically tailored for fig detection in dense foliage has shown improvements over previous models including Faster R-CNN, emphasizing YOLO's capability in challenging conditions[39]. YOLOv8 has also been successfully applied to size immature green apples using geometric shape fitting techniques, showcasing high precision and robustness against occlusions in orchard environments[4]. Furthermore, an enhanced YOLOv8 model has proven to be effective for detecting Yunnan Xiaomila peppers, integrating attention mechanisms and deformable convolutions to handle small targets against complex backgrounds[40]. Additionally, modifications to YOLOv5 have improved plum recognition, adapting the algorithm to handle fruit occlusions and environmental variability[41]. Another study with YOLOv7-based model equipped with attention mechanisms and specialized pooling has shown to achieve high precision in detecting plums in natural settings[42].

In orchard automation, YOLO object detection models have been pivotal in enhancing the accuracy and efficiency of fruit detection[4][43][44], flower identification[45][46][47], and automated harvesting processes[48][49][50]. These models adeptly identify and classify fruits at various stages of ripeness, detect flowers with high precision, and facilitate efficient harvesting operations. The development of YOLO models has introduced significant improvements specifically tailored to meet the challenges faced in agricultural environments. For instance, the introduction of multi-scale predictions in YOLOv5 has improved the detection of small and clustered objects like flowers and young fruits, which are crucial during the early stages of crop yield management[51].

The rapid advancement of YOLO-based models has significantly enhanced their precision and processing speeds, vital for implementing robotic solutions such as fruitlet thinning in orchards. Given these swift innovations, continuous evaluation of newer models is essential to harness these improvements for agricultural automation.

This study provides a comprehensive evaluation of the latest YOLO versions: YOLO11, YOLOv10, YOLOv9, and YOLOv8, targeting fruitlet detection in commercial apple orchards by examining 22 configurations across these models and utilizing a commercial orchard’s dataset of RGB images from an iPhone 14 across four apple varieties: Scifresh, Scilate, Honeycrisp, and Cosmic Crisp. Each image was annotated with a complete count of visible and occluded apples directly observed in the field, providing a comprehensive dataset for validation. The study also validates the top-performing models using machine vision sensor.

The specific contributions of this study are:

- **Model Training and Configuration:** Comprehensive evaluation of the latest YOLO object detection models implemented across 22 configurations: YOLOv8 (5 configurations), YOLOv9 (6 configurations), YOLOv10 (6 configurations), specifically optimized for detecting green fruitlets in commercial apple orchards, and YOLO11 (5 configurations).
- **Comprehensive Metrics:** Detailed examination of detection precision metrices, computational efficiency, and processing speeds at 3 steps (preprocess, inference and post-process) of the deep learning workflow.
- **Validation Across Varieties:** In-field counting accuracy validation using four apple varieties not included in the training set, to test generalizability and robustness of the models under varied agricultural conditions.
- **Integration of Smartphone Technology:** Utilization of high-resolution RGB images from an iPhone 14 Pro Max for adaptability of advanced smartphone imaging in field setting.

The remainder of this paper is organized as follows. First, a detailed background and overview of the YOLO models, specifically YOLOv8, YOLOv9, YOLOv10 and YOLO11, elucidating their architectural differences and enhancements are provided. The subsequent sections delve into the methodology employed in configuring and testing these models, followed by an extensive presentation of results and discussions that highlight key findings and performance metrics. The paper concludes with a discussion on the implications of these results for future work and potential improvements.

2. YOLO Background and Overview: Insights into YOLOv8, YOLOv9, YOLOv10 and YOLO11 Models

Figure 2 shows the timeline history of the YOLO from its release to upto date version as YOLO11. The YOLO object detection algorithm was first introduced by Joseph Redmon et al.[52] in 2015, revolutionized real-time object detection by combining region proposal and classification into a single neural network, significantly reducing computation time. YOLO's unified architecture divides the image into a grid, predicting bounding boxes and class probabilities directly for each cell, enabling end-to-end learning[52]. YOLO is versatile, and its real-time detection capabilities have revolutionized not only agriculture[53], but also many other field such as medical object detection[54], autonomous vehicle industry[55], security and surveillance systems[56], and industrial manufacturing[57] where accuracy and speed are paramount. The current state-of-the-art iteration of YOLO version is YOLOv10, and Figure 2b shows the FPS and mAP of each model starting from YOLOv1 to YOLO11.

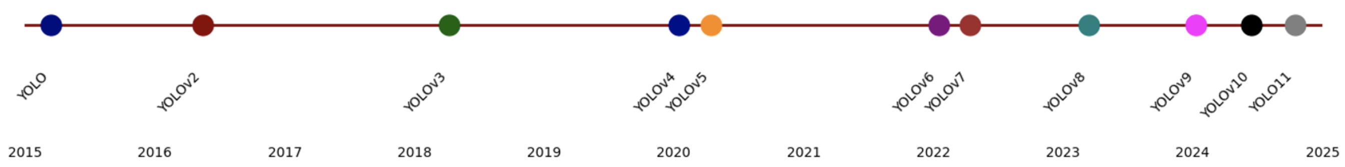


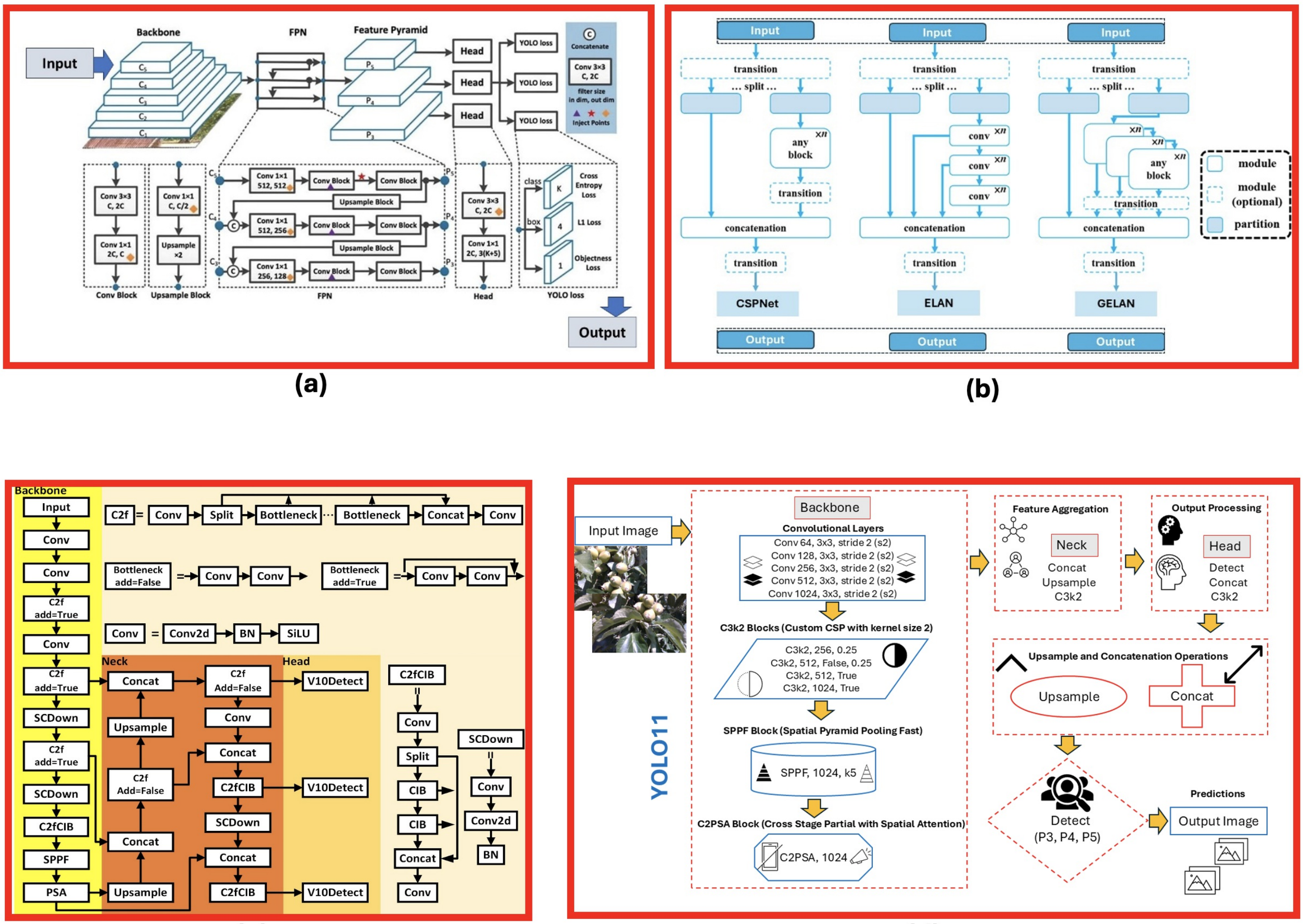
Figure 2. Evolution of YOLO and Performance of YOLO models from 2015-2024: (a) Timeline of YOLO version releases, showcasing advancements from YOLOv1 to YOLO11

After the release of first YOLO version, YOLOv2, or YOLO9000[58][59], expanded on this foundation by improving the resolution at which the system operated and by being capable of detecting over 9000 object categories, thus enhancing its versatility and accuracy. YOLOv3 further advanced these capabilities by implementing multi-scale predictions and a deeper network architecture, which allowed better detection of smaller objects[60]. The series continued to evolve with YOLOv4 and YOLOv5, each introducing more refined techniques and optimizations to improve detection performance (i.e., accuracy and speed) even further[61][62]. YOLOv4 incorporated features like Cross-Stage Partial (CSP) connections and Mosaic data augmentation, while YOLOv5, developed by Ultralytics, brought significant improvements in terms of ease of use and performance, establishing itself as a popular choice in the computer vision community. Subsequent versions, YOLOv6 through YOLO11, have continued to build on this success, focusing on enhancing model scalability, reducing computational demands, and improving real-time performance metrics[63]. Each iteration of the YOLO series has set new benchmarks for object detection capabilities and significantly impacted various application areas, from autonomous driving and traffic monitoring to healthcare and industrial automation[63].

Starting with YOLOv1, the foundational model introduced significant capabilities with an mAP of 63.4%, although it experienced higher latencies[61][63].This was followed by YOLOv2 and YOLOv3, which further improved detection accuracy, achieving mAPs of 76.8% and 57.9%, respectively, at the cost of increased latency. YOLOv4 continued this trend, reaching an mAP of 43.5% and serving as a bridge to more refined models. YOLOv5 emerged as a popular choice, balancing performance and efficiency with a competitive mAP of 50.7% and a latency of 140 ms[64]. The YOLOv6 series, including variants from YOLOv6-N to YOLOv6-L, offered mAP scores ranging from 37.0% to 51.8%, with moderate latencies, marking a significant milestone in optimizing detection speed and accuracy[63]. Lastly, YOLOv7, including the YOLOv7-tiny and standard YOLOv7 models, achieved mAPs of 56.4% and 51.2% respectively, but with significantly higher latencies, indicating a shift towards prioritizing accuracy over speed.

Likewise, YOLOv8 demonstrates commendable performance with mAP scores ranging from 37.3% to 53.9%, and latencies between 6.16 ms to 16.86 ms. While it made significant strides, YOLOv8 falls slightly short in terms of efficiency and accuracy when compared to its successors. Progressing to YOLOv9[65], this iteration includes models like YOLOv9-N, YOLOv9-S, YOLOv9-M, YOLOv9-C, and YOLOv9-X, which achieve mAP scores from 39.5% to 54.4%[65]. Although YOLOv9 matches YOLOv10 in top mAP scores, its latency figures, particularly for YOLOv9-X, are higher, reflecting lesser efficiency than YOLOv10. This discrepancy highlights YOLOv10's advancements in balancing high accuracy with reduced computational demands. The recent variant in the lineup, YOLOv10[66], introduces a range of variants: YOLOv10-N, YOLOv10-S, YOLOv10-M, YOLOv10-B, YOLOv10-L, and YOLOv10-X offering precision scores from 38.5% to 54.4% on the MS-COCO dataset.

YOLO11 (Glenn Jocher 2024)represents the latest advancement in the YOLO family, building on the strengths of its predecessors while introducing innovative features and optimizations aimed at enhancing performance across various computer vision tasks. Specifically, YOLO11 architecture as shown in Figure 3(d) optimized feature extraction capabilities, enabling it to capture intricate details in images. This model supports a range of applications, including real-time object detection, instance segmentation, pose estimation, and which allows it to accurately identify objects regardless of their orientation, scale, or size., making it versatile for industries like agriculture and surveillance. The YOLO11 model utilizes enhanced training techniques that have led to improved results on benchmark datasets. Notably, as depicted in Figure 3 (e), YOLO11m achieved a mean Average Precision (mAP) score of 95.0% on the COCO dataset while utilizing 22% fewer parameters compared to YOLOv8m, demonstrating greater efficiency without compromising accuracy. With an average inference speed that is 2% faster than YOLOv10, YOLO11 is optimized for real-time applications, ensuring quick processing even in demanding environments. These specifications position YOLO11 as a powerful tool for advancing AI applications, particularly in sectors where rapid and accurate image analysis is crucial.

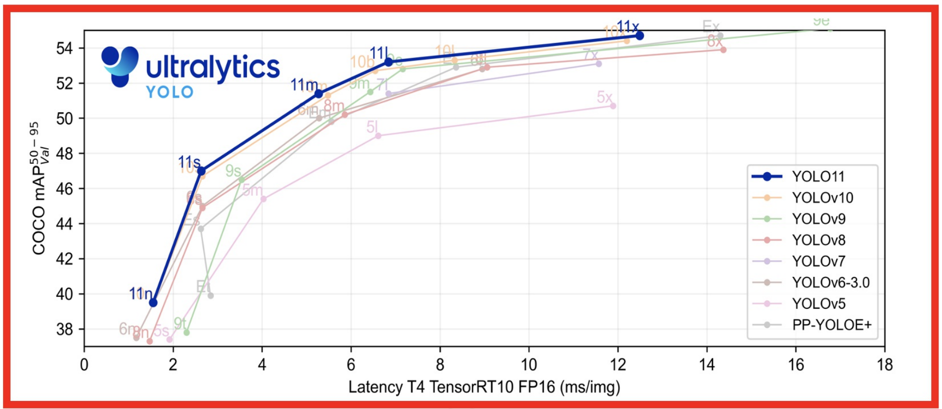


(a)

(b)

(c)

(d)



(e)

Figure 3. Architecture diagrams of YOLOv8, YOLOv9 and YOLOv10 object detection algorithms: (a) YOLOv8 architecture integrates a convolutional backbone and a Feature Pyramid Network for enhanced multi-scale detection ; (b) YOLOv9 architecture incorporates CSPNet, ELAN, and GELAN modules to optimize feature integration and computational efficiency; (c) YOLOv10 architecture advances with a dual label assignment strategy and a Path Aggregation Network, improving precision in object localization and classification; and (d) YOLO11 architecture for immature green fruit detection and; (e) Performance results on benchmark datasets over YOLOv10, YOLOv9, YOLOv8, YOLOv7, YOLOv6, YOLOv5 and PP-YOLOE+

3. Methods

This study was conducted across commercial apple orchards in Prosser and Naches, Washington State, USA, focusing on evaluating datasets of apple fruitlets before thinning from four varieties: Scifresh, Scilate, Cosmic Crisp, and Honeycrisp as described by Figure 4. RGB images were acquired using a machine vision sensor IntelRealsense 435i (Intel Corporation, California, USA). These images were then manually labeled to prepare consistent datasets for training all configurations of YOLOv8, YOLOv9, and YOLOv10 models as illustrated by Figure 4. A total of 22 model configurations were examined: 5 for YOLOv8, 6 for YOLOv9, 6 for YOLOv10, and 5 for YOLO11 each trained under standardized hyperparameter settings on the same computational system to ensure consistency and comparability throughout the process. The trained models' performance was then evaluated using the prepared datasets, followed by in-field counting validation using additional RGB images captured with an Apple iPhone 14 Pro Max smartphone (Apple Inc., California, USA). This validation process aimed to assess each model’s fruit counting accuracy against manually counted ground truths, which included scenarios with occluded apples. Furthermore, this validation step was designed to rigorously test the highest-performing models in terms of speed and accuracy using images from different sensors. The details of this comprehensive methodology are illustrated in Figure 4, providing a visual reference for the experimental setup and data collection approach employed in this study.

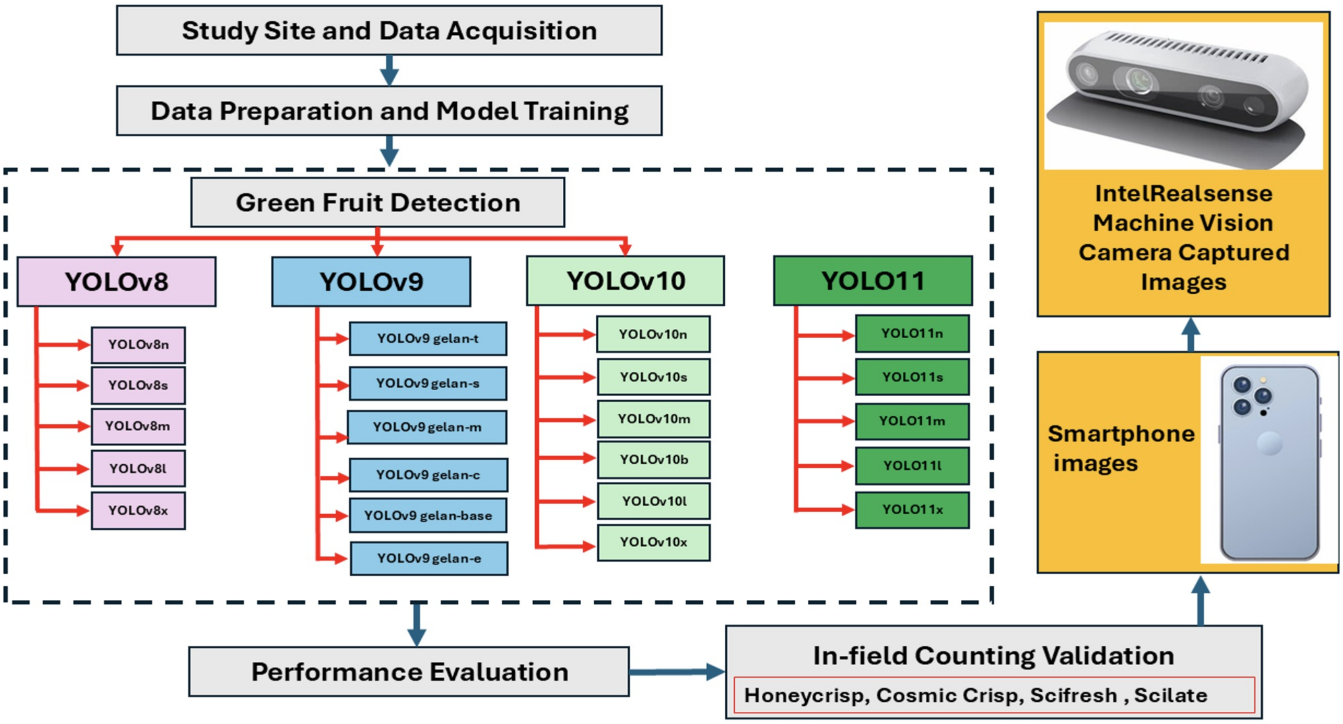


Figure 4. Flow diagram of this study of comparison of YOLOv8, YOLOv9, YOLOv10, and YOLO11 illustrating the study's methodology, including data collection, model training, and validation across multiple sensors and apple varieties in commercial orchards

3.1. Study Site and Data Acquisition

The image data collection for this study was conducted at Allan Brothers Orchard in Prosser, Washington State, USA, focusing on datasets of immature green apples for training models. RGB images were acquired using an Intel RealSense 435i camera during the month of June 2024. For validation, additional datasets were collected from different locations and times: Cosmic Crisp apple images from the ROZA experiment station at WSU IAREC in Prosser, Honeycrisp apple images from an orchard in Naches owned by Allan Brothers Fruit Company. Furthermore, validation of apple counts using the Intel RealSense camera was conducted in June 2023 in Scifresh apple orchard. All images were captured prior to the fruitlet thinning process conducted by orchard workers. The details of each machine vision sensor used in data collection of this study are:

Intel RealSense D435i: This camera features a 2-megapixel RGB sensor capable of capturing high-quality images. Operating at a resolution of 1280×720 pixels, it provides a comprehensive view with a 69.4° horizontal and 42.5° vertical field-of-view, ensuring broad coverage in various environments. The Intel RealSense D435i is designed to be compact and lightweight, making it highly effective for capturing RGB data in diverse settings.

Apple Iphone14 Pro Max: The Apple iPhone 14 Pro Max features an advanced 48-megapixel RGB camera featuring a 24 mm lens with an f/1.78 aperture and second-generation sensor-shift optical image stabilization, ensuring high-resolution image capture with enhanced clarity. The camera's field of view spans 120° horizontally and 90° vertically, supported by a seven-element lens for minimized optical aberrations. It offers up to 15x digital zoom and 4K video recording capabilities. Additionally, the integration of Photonic Engine and Smart HDR 4 technologies optimizes performance in low-light conditions and dynamic range, making it ideal for detailed visual data collection in varying lighting environments.

3.2. Data Preparation and Model Training

A total of 1,147 images were manually annotated with bounding boxes using the online labeling platform provided by Roboflow. These images were allocated into training and validation at the ration of 8:2 respectively, facilitated by Roboflow's distribution tools. No image preprocessing steps were undertaken in this study, as the objective was not to enhance any specific model but rather to compare and evaluate the performance of these models using raw data collected in natural orchard settings.

The computational analyses for this study were conducted on a high-performance workstation equipped with an Intel Xeon(R) W-2155 CPU, featuring a base clock speed of 3.30 GHz across 20 cores. This setup provided substantial processing power necessary for handling intensive data processing tasks. The workstation was also outfitted with NVIDIA Corporation GP102 [TITAN Xp] graphics cards, enhancing its ability to perform complex image processing and machine learning tasks efficiently. The system included a substantial storage capacity of 7.0 TB, facilitating extensive data management and analysis. It operated under Ubuntu 20.04.6 LTS, a robust and stable 64-bit operating system, using GNOME version 3.36.8 for its graphical interface and X11 as its windowing system.

The analysis of YOLOv8, YOLOv9, YOLOv10 and YOLO11 encompassed a total of 22 configurations (five for YOLOv8, six for YOLOv9, six for YOLOv10, and five for YOLOv11). Each model configuration was rigorously tested for precision, recall, mAP@0.5, and image processing speed. For the training of all models, a consistent configuration was rigorously maintained to ensure uniformity and comparability across experiments. Each model was trained for 700 epochs, reflecting a substantial duration to adequately learn and adapt to the dataset's complexities. The batch size was set at 8, optimizing the balance between memory usage and processing speed. Images were resized to a uniform resolution of 640x640 pixels to standardize the input data size across all models. The Stochastic Gradient Descent (SGD) optimizer was employed to update model weights effectively, favored for its efficiency in handling large-scale and complex data. Additionally, the configuration threshold for confidence was set at 0.25, and the Intersection over Union (IoU) threshold was maintained at 0.7, criteria chosen to optimize the balance between precision and recall during object detection tasks.

The hyperparameter settings outlined in table 1 provide a detailed framework for optimizing the training of YOLO models in the study. Key parameters such as the initial and final learning rates were set at 0.01, facilitating a controlled adjustment of learning throughout the training process. Momentum was maintained at 0.937 to ensure consistent updates across epochs, while a minimal weight decay of 0.0005 helped prevent overfitting. The training initiated with a warmup phase spanning 3 epochs to stabilize the learning parameters early in the training. Loss adjustments were specifically tuned, with box loss, class loss, and definition loss set at 7.5, 0.5, and 1.5 respectively, to balance the contributions of different components of the loss function. Image augmentation techniques such as hue, saturation, and value adjustments were precisely defined to enhance model robustness under varied lighting conditions typically found in orchard environments. Specific settings for geometric transformations like rotation, translation, and scaling were employed to simulate different orientations and sizes of objects, crucial for improving the model's ability to generalize across diverse scenarios. The table also highlights the use of flipping and mosaic data augmentation to further enrich the training dataset, ensuring comprehensive exposure to potential real-world variations.

Table 1. Hyperparameter Settings for YOLOv8, YOLOv9, YOLOv10 and YOLO11 used in training YOLO models for fruitlet detection in commercial apple orchards

Hyperparameter	Value	Description
Initial Learning Rate *(lr0)*	0.01	Sets the starting learning rate.
Final Learning Rate *(lrf)*	0.01	Determines the learning rate at the end of training.
Momentum	0.937	Controls the momentum for the SGD optimizer
Weight Decay	0.0005	Helps in regularizing and preventing overfitting.
Warmup Epochs	3.0	Number of initial epochs for learning rate stabilization.
Box Loss Gain *(box)*	7.5	Weight of the bounding box loss component.
Class Loss Gain *(cls)*	0.5	Weight of the class prediction loss component.
Definition Loss Gain *(dfl)*	1.5	Weight of the definition loss component

3.3. Performance Evaluation

To evaluate the detection capabilities of each configuration of YOLOv8, YOLOv9, YOLOv10, and YOLO11, the metrics of precision, recall, and mean Average Precision at Intersection over Union (IoU) threshold of 0.50 (mAP@50) were employed. Precision was calculated as the ratio of true positive detections to the total predicted positives, given by equation 1, where TP denotes true positives and FP denotes false positives. Recall measured the ratio of true positive detections to the actual positives, formulated as equation 2 where FN represents false negatives. The mAP@50 was determined by averaging the precision across all recall levels for an IoU > 0.50. Additionally, the image processing speed for each model was analyzed and compared across three categories: preprocessing, inference, and postprocessing. These evaluations were systematically conducted across 22 configurations of the YOLO models: YOLOv8m, YOLOv8s, YOLOv8l, YOLOv8x, YOLOv8c for YOLOv8; YOLOv9 Gelan-e, YOLOv9 Gelan-c, YOLOv9 Gelan-s, YOLOv9 Gelan-t, YOLOv9 Gelan-m, and YOLOv9 Gelan for YOLOv9, YOLOv10m, YOLOv10s, YOLOv10l, YOLOv10x, YOLOv10c, YOLOv10d for YOLOv10, and YOLO11n, YOLO11s, YOLO11m, YOLO11l, and YOLO11x for YOLO11 to evaluate their efficiency in detecting fruitlets in commercial orchards.

$$\text{Precision} = \frac{TP}{TP + FP} \qquad \text{(Equation 1)}$$

$$\text{Recall} = \frac{TP}{TP + FN} \qquad \text{(Equation 2)}$$

Additionally, the assessment involved analyzing preprocessing, inference, and postprocessing speeds for each model configuration, as these metrics are critical for real-time object detection systems. Preprocessing speed determines how quickly a model can prepare images for detection, inference speed measures the time taken to identify objects within images, and postprocessing speed reflects how swiftly the model finalizes the outputs after detection. Each of these stages is essential for efficient operation in agricultural applications, where timely and accurate detection can significantly impact decision-making and resource management. The computational efficiency of these models directly influences their practical utility in automated fruit detection systems, making this evaluation crucial for advancing agricultural technology solutions.

3.4. In-Field Counting Validation

Upon evaluating the performance metrics of each YOLOv8, YOLOv9, YOLOv10, and YOLO11 configuration for detecting fruitlets, the most effective model from each version was selected based on the highest accuracy achieved at mAP@0.5. These top-performing models from YOLOv8, YOLOv9, YOLOv10, and YOLO11 were further validated to assess their detection capabilities in a real commercial orchard setting across four distinct apple varieties in Washington State. This validation process utilized images collected both by smartphone and machine vision sensor. Initially, images of Scifresh, Scilate, Cosmic Crisp, and Honeycrisp apples were captured using the smartphone. A total of 128 images, 32 per variety, were analyzed to evaluate the models' counting accuracy before the thinning process.

Subsequently, an additional set of images was used for further validation. This included 32 images of Scifresh apples taken with an IntelRealsense machine vision camera. Notably, while the IntelRealsense camera images of Scifresh and Scilate apples served as the training dataset, validation was performed on varieties: Cosmic Crisp and Honeycrisp, which were not included in the training phase. For each image, a count of all visible and occluded apples fruitlets were manually performed directly in the field. This comprehensive manual counting provided a precise ground truth for each image sample across the different apple varieties, ensuring robust validation of the models' performance.

In our study assessing the performance of YOLO models for detecting fruitlets in commercial orchards, Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) were employed as crucial metrics to quantify the accuracy of fruit counts predicted by the models compared to manually counted ground truths. RMSE was calculated using the equation 3:

$$\text{RMSE} = \sqrt{\left(\frac{1}{n}\sum_{i=1}^{n}\left(\text{predicted}_i - \text{actual}_i\right)^2\right)} \qquad \text{(Equation 3)}$$

Here, $predicted_i$ denotes the number of fruits counted by the model,$actual_i$ represents the manually counted fruits for each image, and n is the total number of images. This measure calculated the square root of the average of the squared differences between predicted and actual counts, emphasizing larger errors by squaring the discrepancies, which made it particularly relevant for highlighting significant deviations in model performance.

Likewise, MAE was determined using equation 4:

$$\text{MAE} = \left(\frac{1}{\text{n}}\right) * \sum_{i=1}^{n} \mid \text{predicted}_{\text{i}} - \text{actual}_{\text{i}} \mid \qquad \text{(Equation 4)}$$

Where, *predicted$_i$* and *actual$_i$* denotes the retain their definitions, with MAE calculating the average of the absolute differences between the predicted and actual counts. This metric, being less sensitive to large errors compared to RMSE, provided a straightforward indication of the average error magnitude per image, offering an intuitive measure of prediction accuracy across the dataset.

4. Results and Discussion

The evaluation of performance across different YOLO model configurations: YOLOv8, YOLOv9, YOLOv10, and YOLO11 is comprehensively illustrated in Figures 5, 6, 7, and 8 respectively. Figure 5(a) presents an original image showcasing green apples in a complex orchard environment characterized by shadow and partial sunlight, set against an all-green background. This setting represents a challenging scenario for object detection due to the camouflaging color palette and varying light conditions. Figure 5(b) details the performance of the YOLOv8l model configuration, where a false detection has occurred. Notably, a double bounding box is erroneously placed over a region obscured by trellis wire and leaves, mistakenly identified as fruitlets. This misidentification is highlighted by a yellow circle, indicating the area of false positive detection. Figure 5(c) illustrates the results from the YOLOv8m configuration, which also resulted in a double detection over a single green apple. This overprediction is marked by a yellow dotted circle, showcasing the model's sensitivity to overlapping features within the detection area. Figure 5(d) captures the detection outcomes from the YOLOv8x model, which successfully identified five apples accurately in the described complex conditions. This model's effectiveness in handling the intricacies of the environment underscores its robustness. Figures 5(e) and 5(f) depict the performances of the YOLOv8n and YOLOv8s configurations, respectively. Both configurations have correctly identified the green apples, demonstrating their capability to accurately detect objects without the interference observed in other configurations.

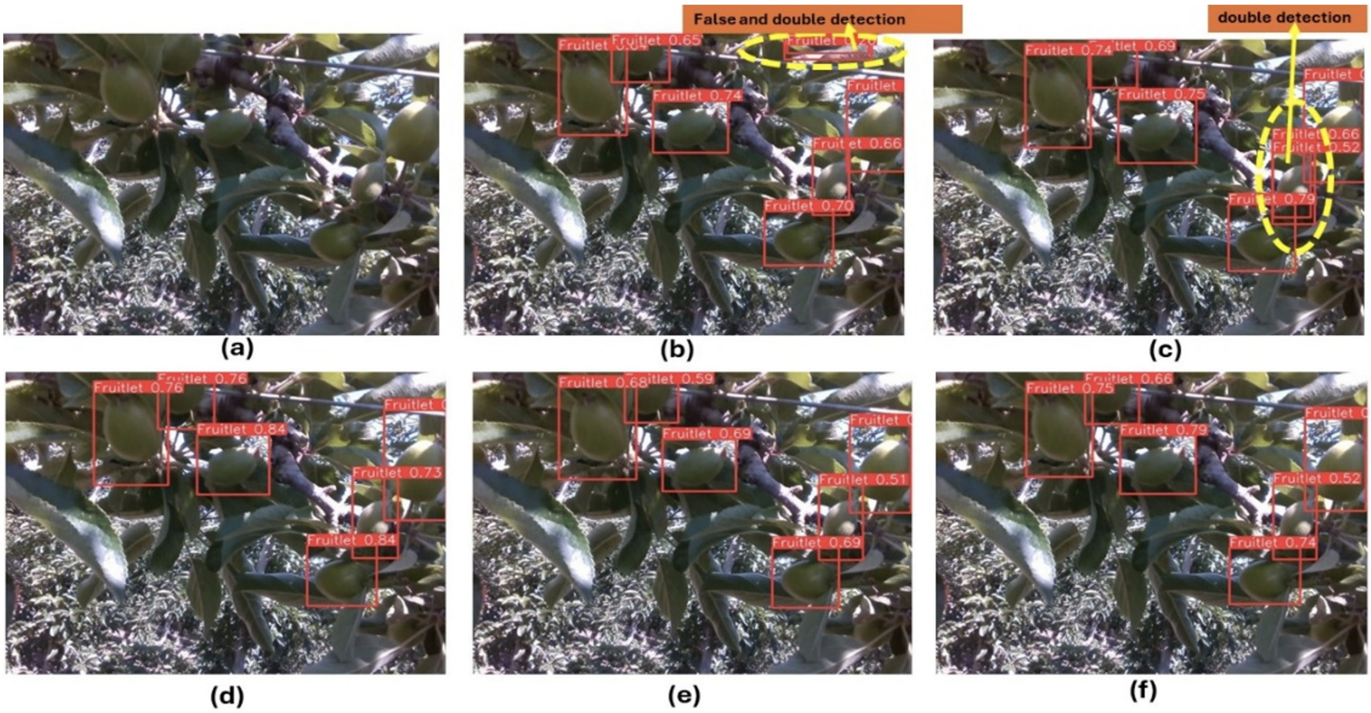


Figure 5. Illustration of comparative performance of YOLOv8 object detection algorithm for fruitlet detection in a complex orchard environment: a) Original Image; b) YOLOv8l detection results; c) YOLOv8m detection results; d) YOLOv8n detection results ; e) YOLOv8s detection results; and e) YOLOv8x detection results

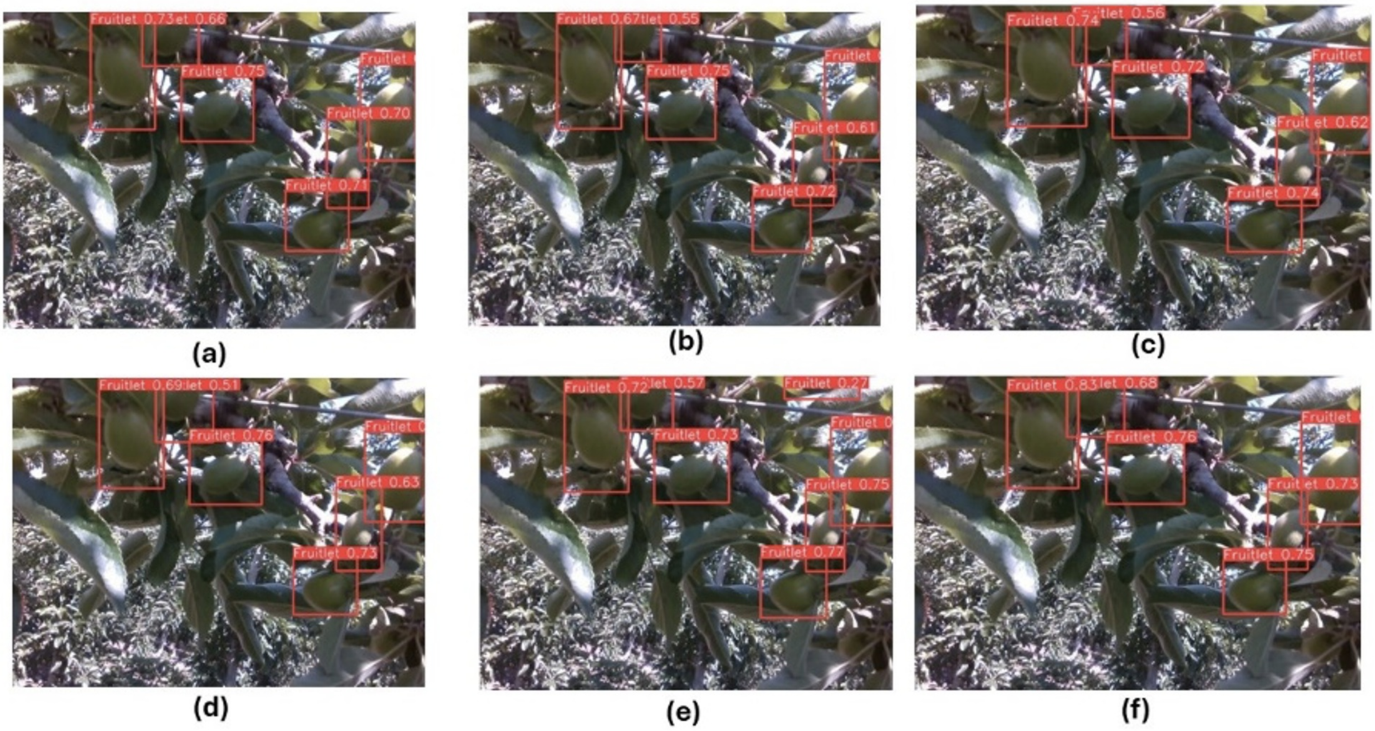


Figure 6. Illustration of comparative Performance of YOLOv9 object detection algorithm for fruitlet detection in a complex orchard environment: a) YOLOv9 gelan-c detection results; b) YOLOv9 gelan-e detection results; c) YOLOv9 gelan-s detection results; d) YOLOv9 gelan-t detection results; e) YOLOv9 gelan-m detection results; and f) YOLOv9 gelan base detection results.

Likewise, Figure 6 illustrates the detection performance of various YOLOv9 configurations across a challenging orchard environment, comparing them against earlier YOLOv8 results. The configurations YOLOv9 Gelan-c (subfigure a), YOLOv9 Gelan-e (subfigure b), YOLOv9 Gelan-s (subfigure c), YOLOv9 Gelan-t (subfigure d), and YOLOv9 Gelan base (subfigure f) all demonstrated excellent accuracy in identifying fruitlets, with no false detections noted. However, an exception was observed in YOLOv9 Gelan-m (subfigure e), which uniquely recorded a false positive.

Figure 7 showcases the detection capabilities of the YOLOv10 object detection algorithm on the same original image previously analyzed in Figures 5 and 6. Subfigure 7(a) highlights a scenario where YOLOv10b mirrors the false detection previously observed with YOLOv8l (Figure 5b), incorrectly identifying a non-fruit area as containing fruitlet with a single bounding box. This misidentification underscores potential challenges in distinguishing complex background elements from target objects under certain conditions.

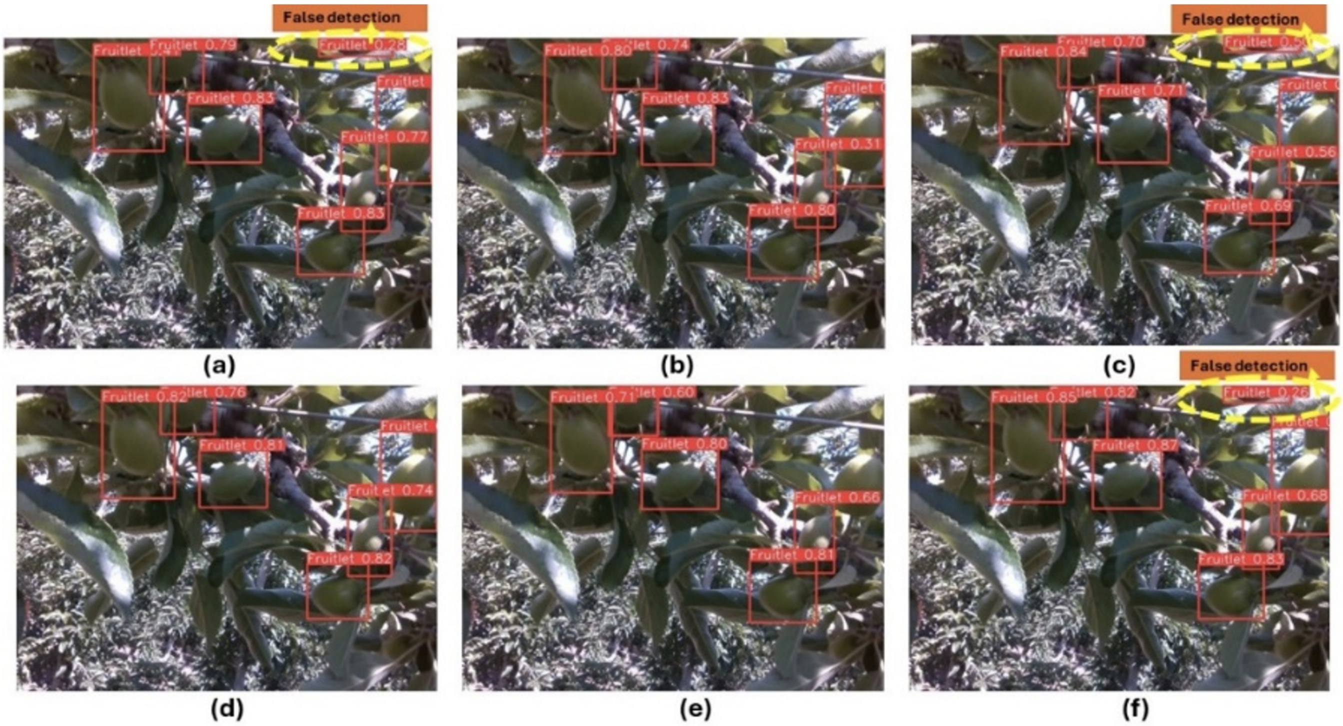


Figure 7. Examples of YOLOv10 object detection algorithm's all configurations for fruitlet detection in a complex orchard environment: (a) YOLOv10b incorrectly detects a non-fruit area as containing fruitlet; (b) YOLOv10l accurately identifies fruitlets within the orchard. (c) YOLOv10m falsely detects non-existent green apples. (d) YOLOv10-N demonstrates precise fruitlet detection. (e) YOLOv10s successfully identifies fruitlets with high accuracy. (f) YOLOv10x also incorrectly identifies non-existent fruitlets, mirroring the error of YOLOv10b

Figure 8 presents examples of YOLO11 detection outcomes, demonstrating the model's exceptional performance in identifying nearly all green apple fruitlets. Notably, these results are attributed to the YOLO11n configuration, which recorded the fastest inference speed of 2.4 ms among the 22 configurations examined in this study, spanning variants from YOLOv8 to YOLO11. Despite the challenges posed by densely clustered environments, as depicted in Figure 8a, the YOLO11 object detection algorithm achieved near-perfect detection rates for all visibly observable apple fruitlets. Furthermore, Figure 8b illustrates the model's capability to detect partially visible, occluded fruitlets located at the edges of the camera frame, underscoring YOLO11's effectiveness in complex commercial orchard scenarios. However, Figure 8b also highlights a limitation in the yellow marked area where the model failed to detect an apple fruitlet, likely due to shadowing from dark lighting conditions. This issue suggests that enhancements could be made by expanding the training dataset. Given that this study was conducted using a research-level dataset comprising approximately a thousand images, scaling up the dataset is anticipated to address these detection challenges in broader industrial and robotic applications.

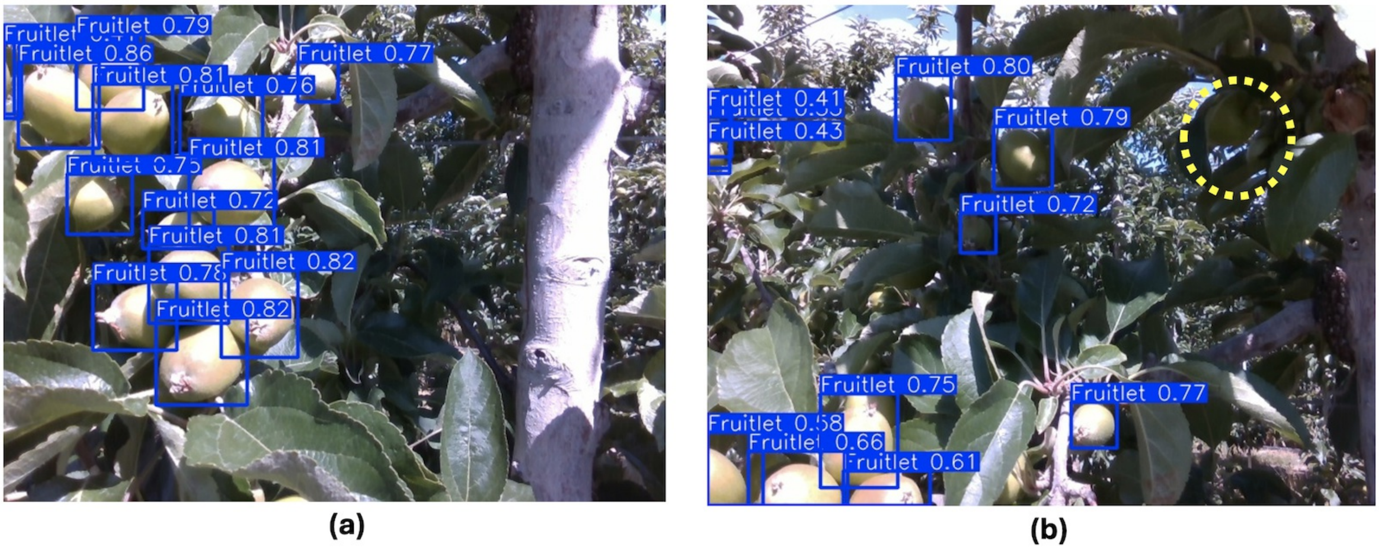


Figure 8. Examples of YOLO11-based fruitlet detection in complex orchard environments: (a) YOLO11n model successfully identifies nearly all visible green apple fruitlets in densely clustered settings. (b) YOLO11n detects partially occluded fruitlets at the frame's edge, with some missed due to shadowing.

4.1. Assessment of Detection Accuracy: Precision and Recall Metrics

The comparative analysis of detection accuracy across different YOLO configurations, as illustrated in Figure 9. Specifically, in terms of precision, YOLOv10x emerged as the top performer among all the model configurations, achieving the highest precision score of 0.891. This indicates its superior capability to accurately identify targets with minimal false positives, thereby excelling in precise object detection scenarios. Conversely, when considering recall, YOLOv9 Gelan-m distinguished itself by achieving the highest recall value of 0.899. This model excels in capturing the highest proportion of true positives, demonstrating its effectiveness in comprehensive detection tasks, even at the risk of slightly increasing false positives.

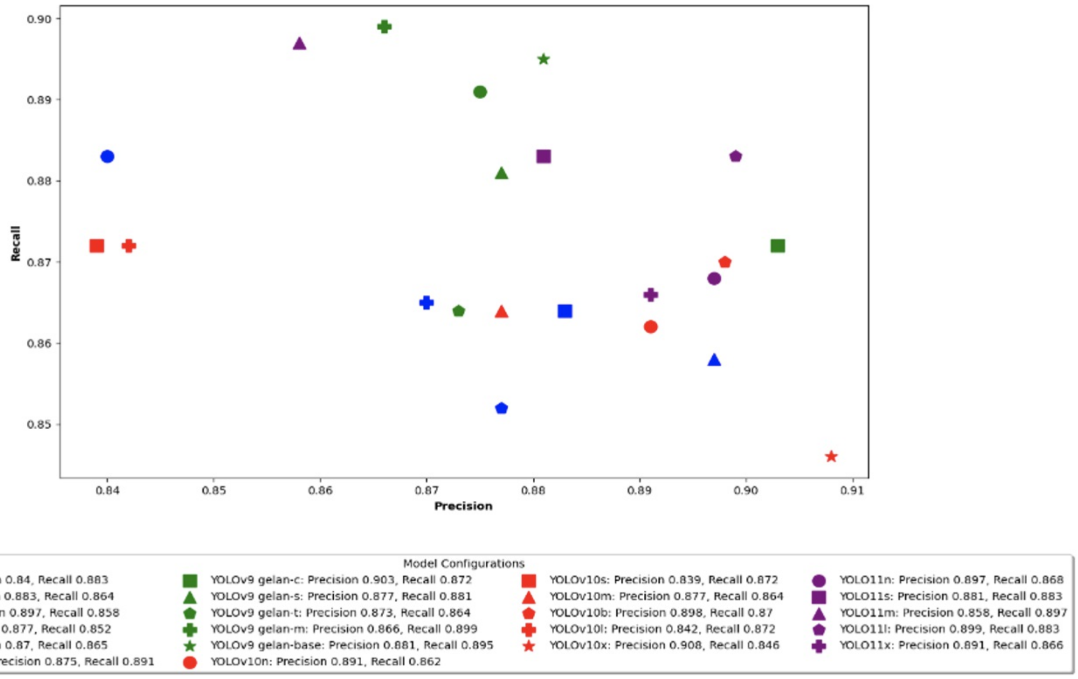


Figure 9. Scatter plot illustrating precision and recall for all configurations of YOLOv8, YOLOv9, YOLOv10, and YOLO11 object detection algorithms for fruitlet detection in complex and commercial apple orchards.

The analysis also highlighted the performances of other models in each series. For instance, YOLOv9 Gelan-c was notable for its impressive precision of 0.903, underscoring its accuracy in fruitlet detection while minimizing the misidentification of unrelated objects. Within the YOLOv8 series, YOLOv8m stood out with a precision of 0.897, indicating effective and accurate target identification. Further examination revealed that YOLO11 models, such as YOLO11l, also demonstrated robust performance with a blend of high precision and recall, recording scores of 0.899 and 0.883 respectively. These models enhance the detection accuracy in complex environments, affirming the advancements in YOLO technology. Overall, the comparative analysis from Figure 9 clearly illustrates that YOLOv10x and YOLOv9 Gelan-m lead in precision and recall respectively, reflecting ongoing enhancements in the YOLO series that cater to diverse and challenging detection environments.

Figure 10 showcases the performance metrics of the best-performing YOLO configurations for detecting fruitlets, segmented into subfigures for clarity. Subfigures 10(a), 10(b), and 10(c) illustrate the precision, recall, and F1-score, respectively, for YOLOv8n, which outperformed all other YOLOv8 configurations. Subfigures 10(d), 10(e), and 10(f) detail the superior performance of YOLOv9 Gelan-e, the best among the six configurations of YOLOv9 tested. Furthermore, Subfigures 10(g), 10(h), and 10(i) present the performance metrics for YOLOv10n, which emerged as the leading configuration in the YOLOv10 series. Finally, subfigures 10(j), 10(k), and (l) present the performance metrics for YOL11n as a demonstration of best configurations out of each model from YOLO11 family.

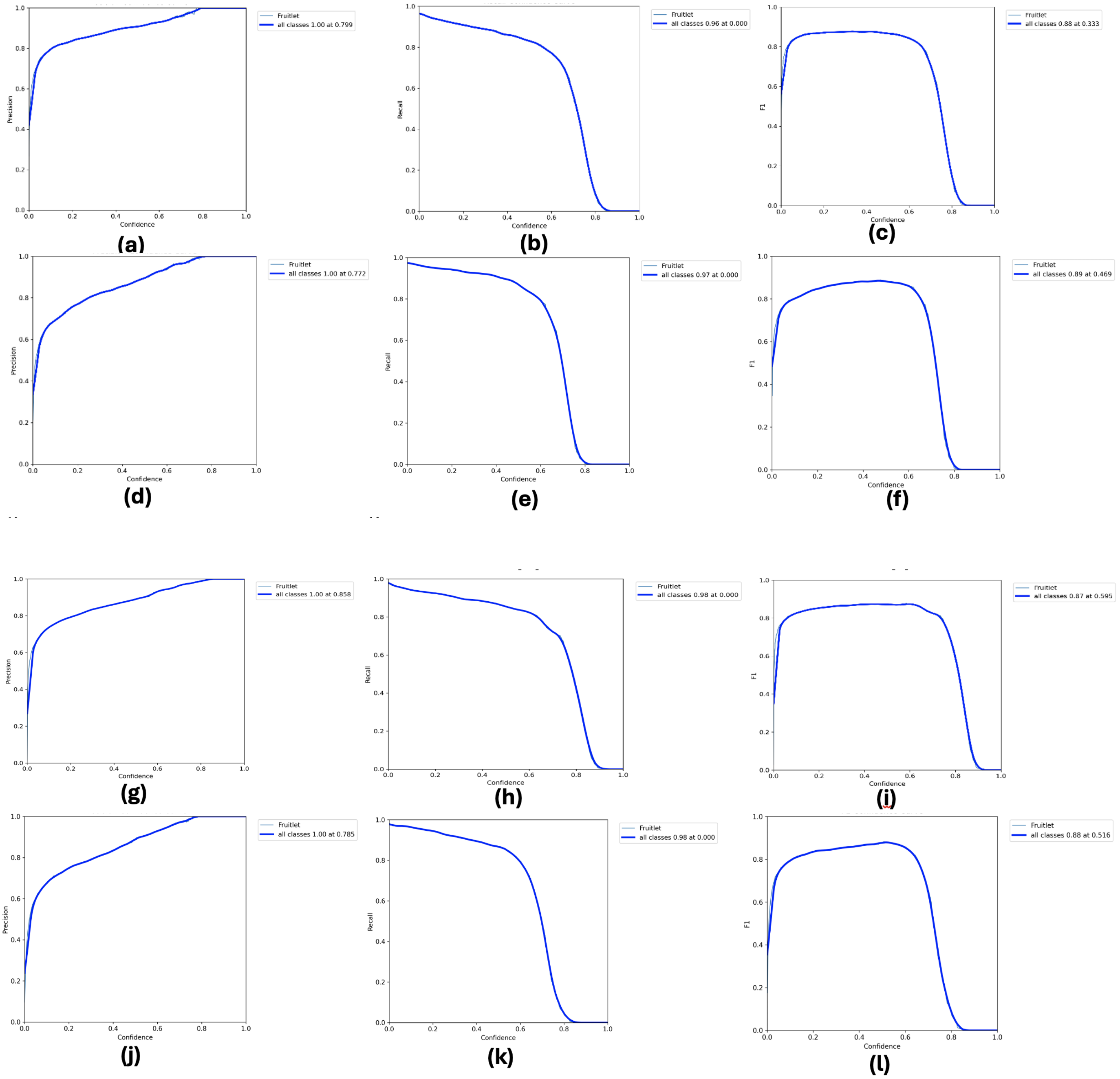


Figure 10. Performance Metrics for the best YOLO Models out of each configuration across YOLOv8, YOLOv9, and YOLOv10 on Fruitlet Detection in complex orchard environments a) YOLOv8n precision confidence curve; b) YOLOv8n recall confidence curve; c) YOLOv8n F1-score curve; d) YOLOv9 gelan-e precision confidence curve; e) YOLOv9 gelan-e recall confidence curve; f) YOLOv9 gelan-e F1-score curve; g) YOLOv10n precision confidence curve; h) YOLOv10n recall confidence curve; i) YOLOv10n F1-score curve; j) YOLO1n precision confidence curve; k) YOLO11n recall confidence curve; l) YOLO11 F1-score curve

4.2. Evaluation of Detection Consistency: Mean Average Precision at IoU=0.50

The assessment of mean Average Precision at an Intersection over Union (IoU) threshold of 0.50 (mAP@0.50) across various YOLO model configurations in Figure 11 provides profound insights into their efficacy in detecting fruitlets within agricultural settings. Among all evaluated configurations, YOLOv9 models, particularly YOLOv9 Gelan-e and YOLOv9 Gelan-base, stand out with the highest mAP@0.50 scores of 0.935 both, respectively. However, all configurations of

YOLOv9 achieved higher mAP@50 as depicted in Figure 10. These scores not only exceed those achieved by all configurations of YOLOv8 and YOLOv10 but also underscore YOLOv9's superior precision in object detection tasks.

The newly included YOLO11 series introduces impressive scores as well: YOLO11n records an mAP@0.50 of 0.926, YOLO11s at 0.933, YOLO11m at 0.924, YOLO11l at 0.932, and YOLO11x at 0.922, indicating enhanced detection capabilities across these newer configurations. Within the YOLOv10 series, YOLOv10n leads with an mAP@0.50 of 0.921, closely followed by YOLOv10b and YOLOv10-M, which record scores of 0.919 and 0.917 respectively. Despite their strong performance, these figures remain slightly below the peak values presented by YOLOv9, indicating that specific enhancements in YOLOv9 have likely boosted its accuracy capabilities. In contrast, the YOLOv8 configurations show a wider spectrum of performance, with YOLOv8s achieving the highest mAP@0.50 within its group at 0.924, nearly matching the top-performing configurations of YOLOv10. Nonetheless, configurations such as YOLOv8l and YOLOv8x, with lower scores of 0.912 and 0.914 respectively. This variation may be attributed to an over-parameterization of the models for the given task. Considering the dataset's simplicity and fewer categories relative to the more complex COCO dataset, the extensive parameters in these models might be excessive and potentially obstruct optimal performance.

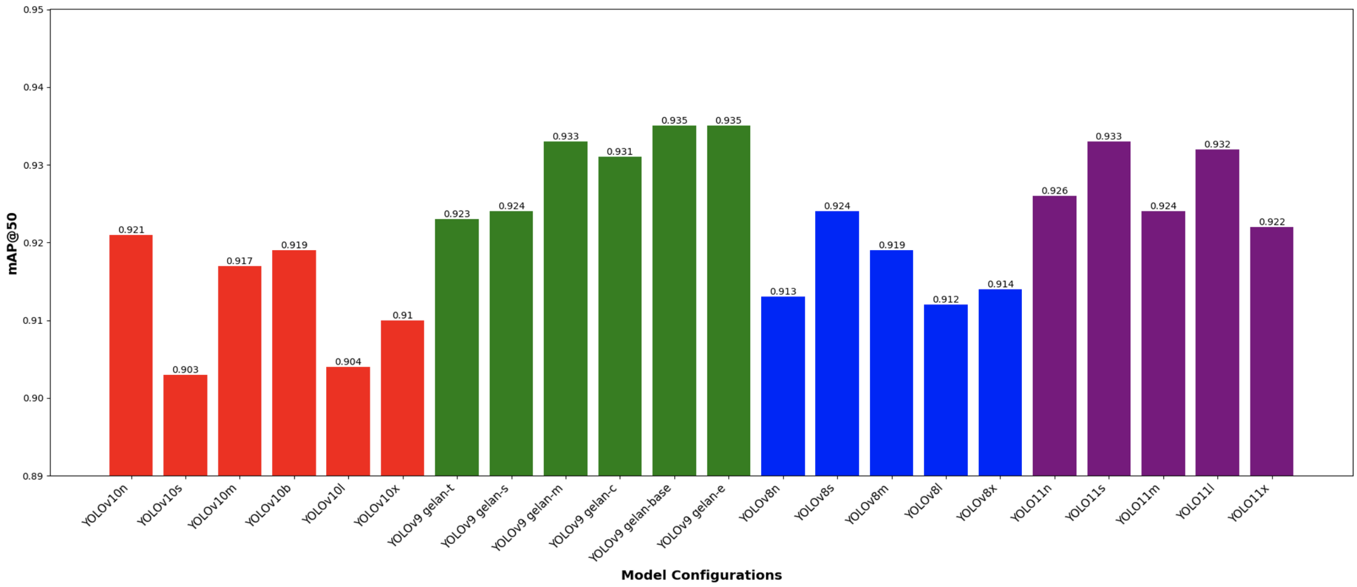


Figure 11. Bar diagram showing mAP@50 scores for all 17 Configurations of YOLOv8, YOLOv9, YOLOv10, and YOLO11 for fruitlet detection in commercial orchards.

4.3. Analysis of Computational Efficiency: Image Processing Speed

In assessing computational efficiency, particularly image processing speeds, YOLO11 emerged as the top performer, achieving remarkably low inference speeds across its variants, with an exceptional rate of just 2.4 ms. Moreover, the preprocessing speeds for YOLO11 were notably quick, registering at 0.1 ms for all variants, except for YOLO11l, which still performed impressively at 0.2 ms. This rapid preprocessing facilitates faster readiness of images for subsequent detection phases. Despite YOLO11's superior performance, YOLOv8x was also notable within its series for achieving the fastest preprocessing speed at merely 0.9 ms. While preprocessing speeds are expected to be consistent across models, the discrepancies observed, particularly in YOLOv8x's speed, are likely due to random variations rather than fundamental differences in model architecture.

Expanding on the theme of processing efficiency, YOLOv8 configurations demonstrated excellence in inference speed. YOLOv8n, in particular, recorded a speed of 4.1 ms, significantly outpacing the fastest models from the YOLOv9 and YOLOv10 series. In comparison, the quickest YOLOv9 model, YOLOv9 Gelan-s, logged an inference time of 11.5 ms, and the leading YOLOv10 model, YOLOv10-s, achieved 5.5 ms. These results underscore YOLOv8n’s superior capability in rapid image processing, highlighting its robustness and suitability for scenarios that demand high-speed, accurate object detection. This analysis reveals that while the YOLO11 series leads in low-latency performance, previous iterations like YOLOv8 still maintain competitive advantages, particularly in environments where quick decision-making is crucial. The detailed performance metrics for each model configuration across the YOLOv8, YOLOv9, YOLOv10, and YOLOv11 series are systematically presented in Table 2.

Table 2. Computational Speeds of YOLOv8, YOLOv9, YOLOv10, and YOLO11 Configurations for fruitlet detection in complex orchard environments.

YOLO models	YOLO Configuration	Preprocessing Speed (ms)	Inference Speed (ms)	Postprocessing Speed (ms)
YOLOv8	YOLOv8n	1.3	**4.1**	2.3
	YOLOv8s	1.3	6.4	2.3
	YOLOv8m	1.3	11.2	**2.1**
	YOLOv8l	1.2	18.7	2.2
	YOLOv8x	**0.9**	24.8	2.3
YOLOv9	YOLOv9 Gelan-t	1.3	14.1	2.2
	YOLOv9 Gelan-s	1.3	**11.5**	2.2
	YOLOv9 Gelan-m	1.3	14	2.1
	YOLOv9 Gelan-c	1.3	17	2
	YOLOv9 Gelan-base	1.2	17.2	2
	YOLOv9 Gelan-e	**1.1**	33.5	**1.9**
YOLOv10	YOLOv10n	1.4	**5.5**	**1.6**
	YOLOv10s	1.3	7.7	1.6
	YOLOv10m	1.3	13	1.6
	YOLOv10b	1.3	16.7	1.5
	YOLOv10l	1.4	19.6	1.5
	YOLOv10x	**1.2**	26.5	1.5
YOLO11	YOLO11n	**0.1**	**2.4**	2.2
	YOLO11s	**0.1**	5.0	2.5
	YOLO11m	**0.1**	11.9	**0.6**
	YOLO11l	0.2	14.6	1.2
	YOLO11x	**0.1**	26.3	**0.6**

Note: Preprocessing refers to the initial stage where images are prepared for analysis, involving adjustments such as scaling, normalization, and augmentation to optimize them for detection. Inference refers to the core phase of the process where the model analyzes the preprocessed images to detect and identify objects based on the learned features and patterns. Postprocessing refers to the final stage that refines the outputs from the inference, applying techniques like Non-Maximum Suppression (NMS) to eliminate redundant detections and finalize the list of detected objects.

4.4. Field Validation of Counting Accuracy: RMSE and MAE Metrics

For the counting validation performed on four apple varieties images collected using Apple Iphone 14 smartphone, the top-performing configurations from each YOLO version: YOLOv8n, YOLOv9 Gelan-e, YOLOv10n and YOLO11n were rigorously assessed for accuracy in fruit counting. YOLO11n demonstrated exceptional precision, outperforming prior configurations in RMSE and MAE metrics across various apple varieties. Notably, for Honeycrisp apples, YOLO11n recorded an RMSE of 4.51 and an MAE of 4.07, marking a significant improvement in detection precision. Similarly, for the Cosmic Crisp variety, it achieved an RMSE of 4.59 and an MAE of 3.98. The Scilate variety showed an RMSE of 4.83 and a notably higher MAE of 7.73, while Scifresh apples had an RMSE of 4.96 and an MAE of 3.85. These results indicate YOLO11n's advanced capability in accurately counting and detecting fruitlets under complex orchard conditions.

Despite the strong performance of YOLO11n, the YOLOv9 Gelan-e configuration also showed robust results in previous assessments, particularly excelling with Honeycrisp apples by achieving an RMSE of 4.89 and an MAE of 4.14. For Cosmic Crisp, it registered the lowest RMSE at 4.77 and an MAE of 4.04, followed by RMSEs of 5.54 for Scilate and 5.37 for Scifresh, with corresponding MAEs of 4.16 and 4.23, respectively. These metrics affirm YOLOv9 Gelan-e's effectiveness in precision fruit counting across multiple apple varieties, consistently outperforming YOLOv8 and YOLOv10 configurations. Figure 12 depicts a detailed visual distribution of RMSE and MAE values, capturing in-field counting validation results for the four apple varieties, collected using an Apple iPhone 14 camera during the fruitlet season.

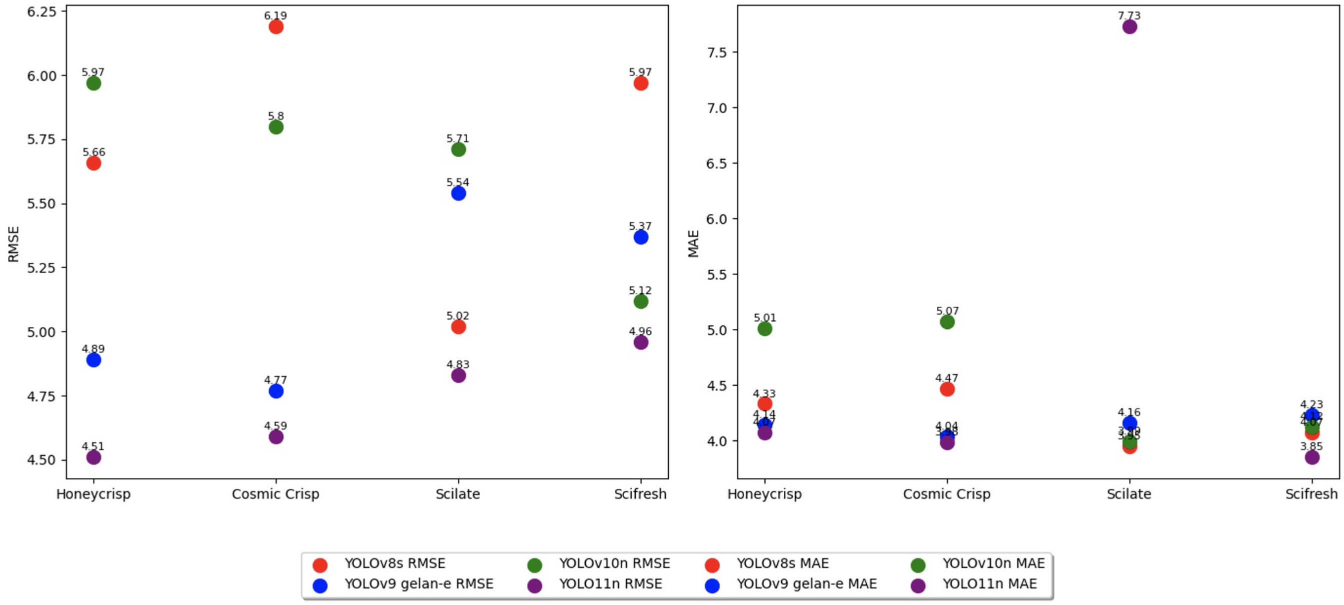


Figure 12. RMSE and MAE for in-field counting validation for Green Apple Detection Using iPhone 14 pro Images: Comparison of RMSE (left) and

Figure 12. RMSE and MAE for in-field counting validation for Green Apple Detection Using iPhone 14 pro images. Comparison of RMSE (left) and MAE (right) for green apple detection across YOLOv8s, YOLOv9 gelan-e, YOLOv10n , YOLO11n configurations, highlighting model accuracy and precision.

In the evaluation of images captured by consumer-scale machine vision cameras for apple counting, the updated results highlight the exceptional accuracy of the YOLO11n configuration. Specifically, YOLO11n demonstrated remarkable performance in counting Scifresh apples, achieving an RMSE of 3.06 and an MAE of 2.33. This outperforms the previously noted accuracy of the YOLOv9 Gelan-e configuration, which led earlier assessments with an RMSE of 3.11 and an MAE of 2.46 for the same variety using Intel Realsense cameras. Figures 12 and 13 illustrate the distribution of RMSE and MAE across multiple configurations and apple varieties; Scilate, Scifresh, Honeycrisp, Cosmic Crisp captured with an Apple iPhone 14, with special attention to the performance on Scifresh apples captured using Intel Realsense cameras.

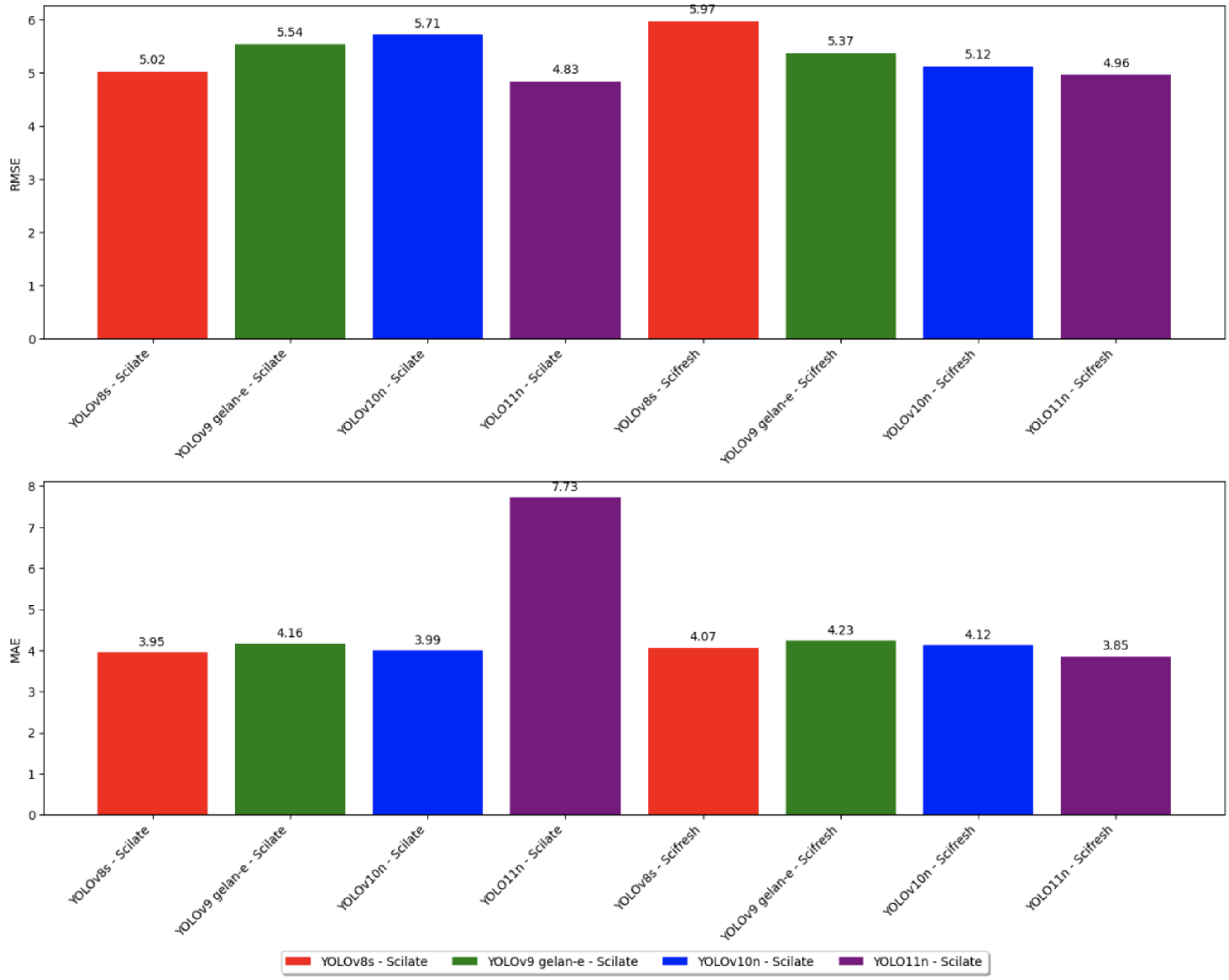


Figure 13. RMSE and MAE for Green Apple Detection Using a consumer grade Machine Vision Sensor Displays RMSE (left) and MAE (right) for green apple detection across YOLOv8s, YOLOv9 gelan-e, YOLOv10n and YOLO11n assessed with Intel Realsense camera

Figure 14 (a-h) demonstrates the proficiency of various YOLO configurations in detecting green apples within different orchard settings, as captured by an Apple iPhone. The sequence of subfigures from 14(a) to 14(d) examines the performance on Scilate apples for YOLOv8s, YOLOv9 Gelan-e, YOLOv10n, and the newly added YOLO11n, respectively. Similarly, subfigures from 14(e) to 14(h) extend this analysis to Scifresh apples. These illustrations affirm the adaptability of each model to distinct orchard architectures and environmental conditions, showcasing their utility in practical agricultural applications. The addition of YOLO11n especially highlights a refined detection capability, evident in its consistent performance across varying scenarios. Figure 15 (a-h) details the detection and counting performance for Honeycrisp and Cosmic Crisp apple varieties.

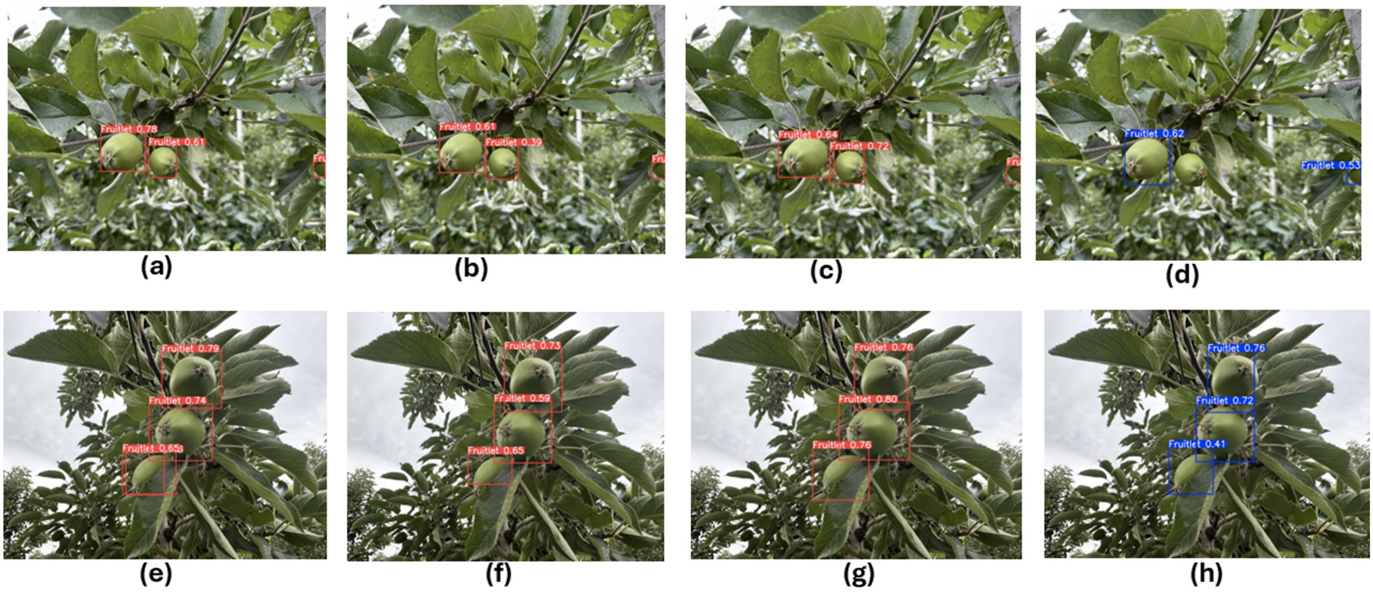


Figure 14. Performance of best YOLO configuration across YOLOv8, YOLOv9 and YOLOv10 Models in fruitlet detection on images collected by Apple Iphone on: 14 a) YOLOv8s on Scilate; b) YOLOv9 Gelan-e on Scilate; c) YOLOv10n on Scilate; d) YOLO11n on Scilate; e) YOLOv8s on Scifresh; f) YOLOv9 gelan-e on Scifresh; g) YOLOv10n on Scifresh; and h) YOLO11n on Scifresh

In the Honeycrisp evaluations, YOLOv8n and YOLOv9 Gelan-c (subfigures 15(a) and 15(b)) demonstrate effective fruit recognition, accurately detecting the majority of apples within a cluster. Subfigure 15(c) reveals a slight misstep by YOLOv10n, where it misidentifies a background element as a fruitlet, pinpointing an area for model improvement. Conversely, the introduction of YOLO11n in subfigure 15(d) showcases superior accuracy with no such errors. For Cosmic Crisp apples, subfigures 15(e) and 15(f) show successful peripheral apple detection by YOLOv8s and YOLOv9 Gelan-e, reflecting robust edge detection capabilities. Subfigure 15(g) highlights a limitation in YOLOv10's ability to recognize an apple in the annotated area. However, YOLO11 (subfigure 15(h)) corrects this oversight by effectively identifying all designated apples, including those in challenging peripheral or obscured positions. Overall, the inclusion of YOLO11 results enriches the comparative study by not only bridging the performance gaps identified in earlier configurations but also by establishing a new benchmark for accuracy in fruitlet detection tasks.

Although the YOLO models demonstrated strong performance in detecting green apples from clusters in commercial orchards across four varieties, as shown in Figures 14 and 15, it is important to note the imaging context. The images were captured using an Apple iPhone 14, which offers high-contrast imaging with enhanced focus and resolution. In contrast, the deep learning models for YOLOv8, YOLOv9, YOLOv10, and YOLO11 were trained on RGB images acquired by IntelRealsense cameras, which differ in resolution, saturation, and overall image quality. This discrepancy highlights the robustness of the models, as they were never trained on the specific varieties shown or on images from the iPhone camera. Despite these differences, the exemplary performance of configurations like YOLOv9 Gelan-e, YOLOv8s, and YOLOv10n showcases their potential for fruitlet detection. This study, involving 1,149 images, suggests that expanding the training datasets and employing more computationally intense environments could further enhance model accuracy and generalizability.

5. Conclusion and Future Suggestions

In this study, we conducted a comprehensive evaluation of various configurations of three state-of-the-art YOLO object detection models (YOLOv8, YOLOv9, YOLOv10, and YOLO11) to assess their effectiveness in detecting green apples before thinning in complex orchard environments using different sensors and conditions. The specific conclusions and future suggestions of this study are specifically summarized in following five points:

- **Model-Specific Performance:** YOLOv9 configurations, particularly YOLOv9 gelan-c and YOLOv9 gelan-m, demonstrated remarkable performance in precision and recall metrics, respectively. YOLOv9 gelan-c achieved the highest precision score of 0.903, showcasing superior accuracy in correctly identifying fruitlets without false positives. Meanwhile, YOLOv9 gelan-m recorded the highest recall rate of 0.899, slightly higher than YOLO11m's 0.897, indicating its efficacy in capturing almost all fruitlets in the images.

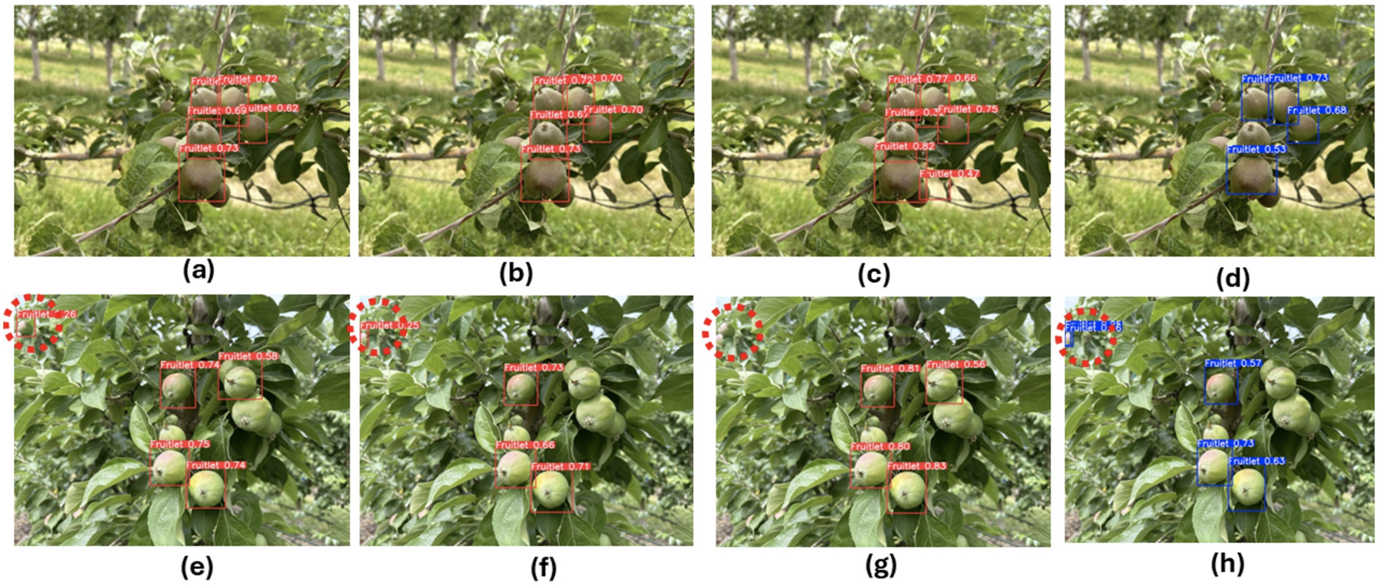


Figure 15. Performance of best YOLO configuration across YOLOv8, YOLOv9 and YOLOv10 Models in fruitlet detection on images collected by Apple Iphone on: a) YOLOv8n on Honeycrisp; b) YOLOv9 gelan-c on Honeycrisp; c) YOLOv10n misidentifies background on Honeycrisp; d) YOLO11n on Honeycrisp ; e)YOLOv8s detects edge apples on Cosmic Crisp; f) YOLOv9 gelan-e on Cosmic Crisp; g) YOLOv10 misses apple on

Cosmic Crisp; and h) YOLO11n on cosmic crisp

- **Accuracy Across Configurations:** In the analysis of mean Average Precision at a 50% Intersection over Union (mAP@0.50), YOLOv9 configurations emerged as leaders, with YOLOv9 Gelan-base and YOLOv9 Gelan-e each securing the highest scores of 0.935. This outstanding performance underscores their robustness in precise object detection within agricultural settings. Notably, the YOLO11a model closely approached these top scores with an mAP@0.50 of 0.933, demonstrating that it is nearly on par with the best-performing models across the YOLOv8, YOLOv9, and YOLOv10 series.
- **Counting Validation:** YOLO11n model demonstrated outstanding precision, achieving the lowest RMSE and MAE across all tested varieties, underscoring its robustness in fruit detection applications.

Specifically, in iphone 14 pro captured images, for YOLO11n recorded the best RMSE scores for each variety: 4.51 for Honeycrisp, 4.59 for Cosmic Crisp, 4.83 for Scilate, and 4.96 for Scifresh. These figures represent the model's consistent ability to accurately estimate fruit counts with minimal error, reflecting its sophisticated detection capabilities. Moreover, the MAE scores further illustrate YOLO11n's precision, with the lowest values recorded at 4.07 for Honeycrisp, 3.98 for Cosmic Crisp, and 3.85 for Scifresh. The exception was Scilate, where YOLO11n recorded a higher MAE of 7.73, suggesting an area for potential.

Likewise for a consumer grade RGB-D camera counting validation, YOLO11n consistently demonstrated the most accurate performance. For Scilate, YOLO11n achieved the lowest RMSE of 4.83 and a significantly higher MAE of 7.73, indicating its precise but inconsistent counting ability under certain conditions. Conversely, for the Scifresh variety, YOLO11n again led with the lowest RMSE of 4.96 and an MAE of 3.85, showcasing its robustness in accurately detecting and counting fruitlets.

- **Sensor-Specific Training Impact:** The study demonstrated that training models on data collected from a specific type of sensor, such as Intel Realsense, significantly enhances model performance when tested with data from the same sensor. This indicates the importance of including diverse sensor data in training phases to ensure robust model performance across various deployment scenarios.
- **Recommendations for Model Deployment:** When deploying YOLO models in automation and robotics, particularly for real-time agricultural tasks such as early-stage crop load management in apple orchards, the choice of model configuration becomes crucial. Each YOLO family has standout configurations optimized for speed and accuracy: YOLOv8n leads its family with an impressive inference speed of 4.1 ms, making it highly suitable for rapid processing needs. In the YOLOv9 series, YOLOv9 Gelan-s excels with a best inference speed of 11.5 ms. For the YOLOv10 configurations, YOLOv10n tops with a speed of 5.5 ms. However, surpassing these, YOLO11n from the YOLO11 series achieves an extraordinary low inference speed of only 2.4 ms, making it the optimal choice for environments where both high speed and precision are critical for efficient automation and real-time decision-making.

Statements and Declarations

Competing interests

The authors declare no competing interests.

Funding

This work was supported by the National Science Foundation and United States Department of Agriculture, National Institute of Food and Agriculture through the ‘‘Artificial Intelligence (AI) Institute for Agriculture’’ Program under Award AWD003473, Zhejiang Provincial Natural Science Foundation (Grant No. LD24E050006, LGN22C130006), and the National Natural Science Foundation of China (Grant No. 32372004).

Acknowledgements

The authors express their heartfelt gratitude to Dave Allan from Allan Brothers Apple Company, which is one of the largest commercial apples growing company over North America, for granting access to the commercial orchard, facilitating this research. Special thanks are also due to Christine Cromar, Bonnie Copeland, Patrick Scharf, Achyut Paudel and Priyanka Upadhayay for their essential support in logistics.

Additionally, we acknowledge the efforts of Astrid Wimmer and Randall Cason for helping in dataset preparation.

Data Availability

All YOLOv8, YOLOv9, YOLOv10, and YOLO11 models can be found at:

- https://docs.ultralytics.com/models/yolov8/
- https://docs.ultralytics.com/models/yolov9/
- https://docs.ultralytics.com/models/yolov10/
- https://docs.ultralytics.com/models/yolo11/

Additional Notes

For further reference on agricultural automation, please see the following publications by Ranjan Sapkota[67][68][69][70]

References

1. ^*M. Mhamed, Z. Zhang, J. Yu, Y. Li, and M. Zhang, “Advances in apple’s automated orchard equipment: A comprehensive research,” Comput Electron Agric, vol. 221, p. 108926, 2024.*
2. ^*F. Xiao, H. Wang, Y. Xu, and R. Zhang, “Fruit detection and recognition based on deep learning for automatic harvesting: an overview and review,” Agronomy, vol. 13, no. 6, p. 1625, 2023.*

3. ^*Q. Zhang, M. Karkee, and A. Tabb, “The use of agricultural robots in orchard management,” in Robotics and automation for improving agriculture, Burleigh Dodds Science Publishing, 2019, pp. 187–214.*
4. a, b, c, d*R. Sapkota, D. Ahmed, M. Churuvija, and M. Karkee, “Immature Green Apple Detection and Sizing in Commercial Orchards using YOLOv8 and Shape Fitting Techniques,” IEEE Access, vol. 12, pp. 43436–43452, 2024.*
5. a, b*F. G. J. Dennis, “The history of fruit thinning,” Plant Growth Regul, vol. 31, pp. 1–16, 2000.*
6. ^*G. Costa, A. Botton, and G. Vizzotto, “Fruit thinning: Advances and trends,” Hortic. Rev, vol. 46, pp. 185–226, 2018.*
7. ^*M. Wei, H. Wang, T. Ma, Q. Ge, Y. Fang, and X. Sun, “Comprehensive utilization of thinned unripe fruits from horticultural crops,” Foods, vol. 10, no. 9, p. 2043, 2021.*
8. ^*M. Shahbandeh, “Most consumed fruits in the U.S. 2021.”*
9. ^*K. Sheth, “Top Apple Producing Countries In The World,” 2018.*
10. ^*USApple, “The Voice of the Apple Industry,” 2021.*
11. a, b*UCDAVIS Gifford Center for Population Studies, “Farm Labor in the 2020s Demand, Supply, and Markets - Report,” https://afop.org/cif/learn-the-facts/.*
12. ^*D. Bochtis, L. Benos, M. Lampridi, V. Marinoudi, S. Pearson, and C. G. Sørensen, “Agricultural workforce crisis in light of the COVID-19 pandemic,” Sustainability, vol. 12, no. 19, p. 8212, 2020.*
13. ^*J. L. Lusk and R. Chandra, “Farmer and farm worker illnesses and deaths from COVID-19 and impacts on agricultural output,” PLoS One, vol. 16, no. 4, p. e0250621, 2021.*
14. ^*V. Marinoudi, C. G. Sørensen, S. Pearson, and D. Bochtis, “Robotics and labour in agriculture. A context consideration,” Biosyst Eng, vol. 184, pp. 111–121, 2019.*
15. ^*J. J. May and T. A. Arcury, “Occupational injury and illness in farmworkers in the eastern United States,” Latinx Farmworkers in the Eastern United States: Health, Safety, and Justice, pp. 41–81, 2020.*
16. ^*G. Earle-Richardson, P. L. Jenkins, D. Strogatz, E. M. Bell, and J. J. May, “Development and initial assessment of objective fatigue measures for apple harvest work,” Appl Ergon, vol. 37, no. 6, pp. 719–727, 2006.*
17. a, b*T. KATAOKA, H. Okamoto, and S. Hata, “Automatic detecting system of apple harvest season for robotic apple harvesting,” in 2001 ASAE Annual Meeting, American Society of Agricultural and Biological Engineers, 1998, p. 1.*
18. ^*J. Zhao, J. Tow, and J. Katupitiya, “On-tree fruit recognition using texture properties and color data,” in 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2005, pp. 263–268.*
19. ^*T. T. Nguyen, K. Vandevoorde, E. Kayacan, J. de Baerdemaeker, and W. Saeys, “Apple detection algorithm for robotic harvesting using a RGB-D camera,” in International Conference of Agricultural Engineering, Zurich, Switzerland, 2014.*
20. ^*J. Wachs, H. I. Stern, T. Burks, V. Alchanatis, and I. Bet-Dagan, “Apple detection in natural tree canopies from multimodal images,” in Proceedings of the 7th European Conference on Precision Agriculture, Wageningen, The Netherlands, 2009, p. 293302.*
21. ^*J. P. Wachs, H. I. Stern, T. Burks, and V. Alchanatis, “Low and high-level visual feature-based apple detection from multi-modal images,” Precis Agric, vol. 11, pp. 717–735, 2010.*

22. ^G. Xuan et al., “Apple detection in natural environment using deep learning algorithms,” IEEE Access, vol. 8, pp. 216772–216780, 2020.

23. ^K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, “Machine learning in agriculture: A review,” Sensors, vol. 18, no. 8, p. 2674, 2018.

24. ^A. Kuznetsova, T. Maleva, and V. Soloviev, “Using YOLOv3 algorithm with pre-and post-processing for apple detection in fruit-harvesting robot,” Agronomy, vol. 10, no. 7, p. 1016, 2020.

25. ^S. Puttemans, Y. Vanbrabant, L. Tits, and T. Goedemé, “Automated visual fruit detection for harvest estimation and robotic harvesting,” in 2016 sixth international conference on image processing theory, tools and applications (IPTA), IEEE, 2016, pp. 1–6.

26. ^G. Xuan et al., “Apple detection in natural environment using deep learning algorithms,” IEEE Access, vol. 8, pp. 216772–216780, 2020.

27. ^H. Kang and C. Chen, “Fast implementation of real-time fruit detection in apple orchards using deep learning,” Comput Electron Agric, vol. 168, p. 105108, 2020.

28. ^S. Sun, M. Jiang, D. He, Y. Long, and H. Song, “Recognition of green apples in an orchard environment by combining the GrabCut model and Ncut algorithm,” Biosyst Eng, vol. 187, pp. 201–213, 2019.

29. ^R. Linker, O. Cohen, and A. Naor, “Determination of the number of green apples in RGB images recorded in orchards,” Comput Electron Agric, vol. 81, pp. 45–57, 2012.

30. ^X. Xia et al., “Detection of young green apples for fruit robot in natural scene.,” Journal of Agricultural Science and Technology (Beijing), vol. 20, no. 5, pp. 64–74, 2018.

31. ^Tian Y, Yang G, Wang Z, Wang H, Li E, Liang Z. "Apple detection during different growth stages in orchards using the improved YOLO-V3 model." Comput Electron Agric. 157: 417–426, 2019.

32. ^Huang Z, Zhang P, Liu R, Li D. "Immature apple detection method based on improved Yolov3." ASP Transactions on Internet of Things. 1(1): 9–13, 2021.

33. ^Wang D, He D. "Channel pruned YOLO V5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning." Biosyst Eng. 210: 271–281, 2021.

34. a, bJia W et al. "An accurate green fruits detection method based on optimized YOLOX-m." Front Plant Sci. 14: 1187734, 2023.

35. ^Zheng Z et al. "A method of green citrus detection in natural environments using a deep convolutional neural network." Front Plant Sci. 12: 705737, 2021.

36. ^Wang J et al. "PG-YOLO: An efficient detection algorithm for pomegranate before fruit thinning." Eng Appl Artif Intell. 134: 108700, 2024.

37. ^Fu X et al. "Green Fruit Detection with a Small Dataset under a Similar Color Background Based on the Improved YOLOv5-AT." Foods. 13(7): 1060, 2024.

38. ^Sun H, Wang B, Xue J. "YOLO-P: An efficient method for pear fast detection in complex orchard picking environment." Front Plant Sci. 13: 1089454, 2023.

39. ^Yijing W, Yi Y, Xue-fen W, Jian C, Xinyun L. "Fig fruit recognition method based on YOLO v4 deep learning." in 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), IEEE, 2021, pp. 303–306.

40. ^Wang F et al. "A lightweight Yunnan Xiaomila detection and pose estimation based on improved YOLOv8." Front Plant Sci. 15: 1421381, 2024.

41. ^Niu Y, Lu M, Liang X, Wu Q, Mu J. "YOLO-plum: A high precision and real-time improved algorithm for plum recognition." PLoS One. 18(7): e0287778, 2023.

42. ^Tang R, Lei Y, Luo B, Zhang J, Mu J. "YOLOv7-Plum: advancing plum fruit detection in natural environments with deep learning." Plants. 12(15): 2883, 2023.

43. ^Chen W, Lu S, Liu B, Chen M, Li G, Qian T. "CitrusYOLO: a algorithm for citrus detection under orchard environment based on YOLOV4." Multimed Tools Appl. 81(22): 31363–31389, 2022.

44. ^Mirhaji H, Soleymani M, Asakereh A, Mehdizadeh SA. "Fruit detection and load estimation of an orange orchard using the YOLO models through simple approaches in different imaging and illumination conditions." Comput Electron Agric. 191: 106533, 2021.

45. ^Wang J, Gao Z, Zhang Y, Zhou J, Wu J, Li P. "Real-time detection and location of potted flowers based on a ZED camera and a YOLO V4-tiny deep learning algorithm." Horticulturae. 8(1): 21, 2021.

46. ^Wu D, Lv S, Jiang M, Song H. "Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments." Comput Electron Agric. 178: 105742, 2020.

47. ^Khanal SR, Sapkota R, Ahmed D, Bhattarai U, Karkee M. "Machine Vision System for Early-stage Apple Flowers and Flower Clusters Detection for Precision Thinning and Pollination." IFAC-PapersOnLine. 56(2): 8914–8919, 2023.

48. ^Junos MH, Mohd Khairuddin AS, Thannirmalai S, Dahari M. "An optimized YOLO-based object detection model for crop harvesting system." IET Image Process. 15(9): 2112–2125, 2021.

49. ^Yijing W, Yi Y, Xue-fen W, Jian C, Xinyun L. "Fig fruit recognition method based on YOLO v4 deep learning." in 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), IEEE, 2021, pp. 303–306.

50. ^Xiao F, Wang H, Xu Y, Zhang R. "Fruit detection and recognition based on deep learning for automatic harvesting: an overview and review." Agronomy. 13(6): 1625, 2023.

51. ^Zhang Y, Li L, Chun C, Wen Y, Xu G. "Multi-scale feature adaptive fusion model for real-time detection in complex citrus orchard environments." Comput Electron Agric. 219: 108836, 2024.

52. a, bRedmon J, Divvala S, Girshick R, Farhadi A. "You only look once: Unified, real-time object detection." in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

53. ^Badgujar CM, Poulose A, Gan H. "Agricultural object detection with You Only Look Once (YOLO) Algorithm: A bibliometric and systematic literature review." Comput Electron Agric. 223: 109090, 2024.

54. ^Ragab MG et al. "A Comprehensive Systematic Review of YOLO for Medical Object Detection (2018 to 2023)." IEEE Access, 2024.

55. ^Dazlee NMAA, Khalil SA, Abdul-Rahman S, Mutalib S. "Object detection for autonomous vehicles with sensor-based technology using yolo." International Journal of Intelligent Systems and Applications in Engineering. 10(1): 129–134, 2022.

56. ^Vijayakumar A, Vairavasundaram S. "Yolo-based object detection models: A review and its applications." Multimed Tools Appl, pp. 1–40, 2024.

57. ^Hussain M. "YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection." Machines. 11(7): 677, 2023.

58. ^Nakahara H, Yonekawa H, Fujii T, Sato S. "A lightweight YOLOv2: A binarized CNN with a parallel support vector regression for an FPGA." in Proceedings of the 2018 ACM/SIGDA International Symposium on field-programmable gate arrays, 2018, pp. 31–40.

59. ^Li R, Yang J. "Improved YOLOv2 object detection model." in 2018 6th international conference on multimedia computing and systems (ICMCS), IEEE, 2018, pp. 1–6.

60. ^Kim KJ, Kim PK, Chung YS, Choi DH. "Performance enhancement of YOLOv3 by adding prediction layers with spatial pyramid pooling for vehicle detection." in 2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS), IEEE, 2018, pp. 1–6.

61. a, bNepal U, Eslamiat H. "Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs." Sensors. 22(2): 464, 2022.

62. ^Mohod N, Agrawal P, Madaan V. "YOLOv4 vs YOLOv5: Object detection on surveillance videos." In: International Conference on Advanced Network Technologies and Intelligent Computing, Springer, 2022, pp. 654–665.

63. a, b, c, dSapkota R, et al. "YOLOv10 to Its Genesis: A Decadal and Comprehensive Review of The You Only Look Once Series." Jun. 2024. doi:10.20944/PREPRINTS202406.1366.V1.

64. ^Li C, et al. "YOLOv6: A single-stage object detection framework for industrial applications." arXiv preprint arXiv:2209.02976, 2022.

65. a, bWang C-Y, Yeh I-H, Liao H-YM. "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information." arXiv preprint arXiv:2402.13616, 2024.

66. ^Wang A, et al. "Yolov10: Real-time end-to-end object detection." arXiv preprint arXiv:2405.14458, 2024.

67. ^Ranjan Sapkota, Rizwan Qureshi. "Multi-Modal LLMs in Agriculture: A Comprehensive Review." 10.36227/techrxiv.172651082.24507804/v1, Sep. 2024.

68. ^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.

69. ^Sapkota R, Ahmed D, Churuvija M, Karkee M. "Immature green apple detection and sizing in commercial orchards using YOLOv8 and shape fitting techniques." IEEE Access. 12: 43436–43452, 2024.

70. ^Khanal SR, Sapkota R, Ahmed D, Bhattarai U, Karkee M. "Machine Vision System for Early-stage Apple Flowers and Flower Clusters Detection for Precision Thinning and Pollination." IFAC-PapersOnLine. 56(2): 8914–8919, 2023.