Object detection method for kiwifruit (*Actinidia deliciosa*) based on improved YOLO11x-mod

Erhan Kahya*

(*Tekirdag Namik Kemal University, Vocational School of Technical Sciences, Department of Computer Technology, Computer Programming Programme, Tekirdag 59030, Turkey*)

Abstract: Kiwi is one of the most important agricultural products in Turkey, and its maturity directly determines the quality and market value of the product. Therefore, there is a need for an accurate and reliable detection system that can minimize post-harvest losses while increasing productivity. In this study, a dataset of 420 images taken under different environmental conditions was expanded to 928 images using data augmentation techniques, and a total of 22 750 kiwi samples were labeled. In the study, the latest deep learning architectures, YOLOv8, YOLOv10, and YOLO11, were systematically compared under identical conditions. As a result of these comparisons, the YOLO11x-mod model showed the highest performance when optimized with hyperparameter strategies such as AdamW optimization, low weight decay, appropriate learning rate, and dropout-mosaic augmentation techniques. This model achieved 81.76% accuracy, 83.97% recognition rate, 86.19% mAP@50, and 66.80% mAP@50:95, delivering excellent results, especially in challenging scenarios such as dense foliage, object overlap, and variable lighting conditions. In addition, the model's operation with 72.5 million parameters, 272 GFLOPs, and 42 fps was found to be suitable for real-time applications, balancing accuracy and speed. The insights gained from the study establish the YOLO11x-mod model as a new reference point for kiwi detection and directly address limitations in the literature regarding insufficient detection of small or overlapping fruits. In this respect, the results of this study not only contribute to science but also show great potential for the integration of the model into robotic harvesting systems and precision agriculture applications.

Keywords: YOLO11x-mod, kiwifruit detection, hyperparameter optimization, object detection, precision farming, robotic systems

DOI: 10.25165/j.ijabe.20261904.10188

Citation: Kahya E. Object detection method for kiwifruit (*Actinidia deliciosa*) based on improved YOLO11x-mod. Int J Agric & Biol Eng, 2026; 19(4): 214–226.

1 Introduction

Kiwifruit (*Actinidia deliciosa*) has become one of the prominent agricultural crops in Turkey. The harvest of the fruit should be planned and controlled depending on kiwifruit maturity. Accurate assessment of fruit maturity is therefore essential for determining the optimal harvest time and maintaining desirable postharvest characteristics.. The harvest period is planned depending on the physical and biochemical characteristics of the kiwifruit. Analysis of the kiwifruit variety "Hayward" grown at different altitudes in Mersin showed that altitude affects the weight and size of the fruit. During harvesting, a gentle technique should be applied to avoid damaging the fruit. Non-destructive acoustic techniques can measure fruit firmness and determine the level of ripeness without damaging the fruit. It is also extremely important to study the nutritional value of kiwifruit before harvest. These studies should be carried out to improve the quality of the fruit and determine the macro- and micronutrient requirements of the kiwifruit plant. Due to the delicate nature of kiwifruit, harvesting must be done manually. Harvesting is a critical stage that affects the quality, shelf life, and even the market price of the fruit. Kiwifruit must be harvested with care, as they are extremely sensitive to damage due to their fragile nature[1]. Mechanical harvesting technology is highly sought after in agriculture to increase productivity, alleviate labor shortages, and ensure consistency of high-quality fruit. Kiwifruit harvesting is particularly challenging because the fruit is soft, the leaves are heavy, and the ripening pattern is irregular[2]. Manual harvesting, although traditionally used, is time-consuming, labor-intensive, and prone to quality variability. Such problems have led to interest in using machine learning models and computer vision methods to develop more efficient automated systems. Object detection, an important aspect of the developed systems, has been revolutionized by deep learning models, making the tools suitable for fruit detection and localization under unfavorable conditions[3].

Deep learning models, particularly the You Only Look Once (YOLO) series, have become leading real-time object detection frameworks. Early uses of YOLO demonstrated the feasibility of using neural networks for object detection at high speeds. As agricultural applications required higher precision and power, subsequent versions such as YOLOv8, YOLOv10, and YOLO11 brought dramatic architectural improvements[4]. These deployments utilized higher-level convolutional layers, more advanced feature extraction mechanisms, and deeper networks, enabling them to tackle challenging tasks such as detecting objects in scenes with overlapping objects, changing light, and decreasing object sizes[5]. These advances have made YOLO models highly attractive for use in agricultural fields such as fruit detection, where flexibility and real-time functionality are of utmost importance.

YOLOv5s-BC was proposed as an artificial intelligence-based

Received date: 2025-09-15 **Accepted date:** 2026-06-17

***Corresponding author: Erhan Kahya**, PhD, research interest: artificial intelligence, deep learning, computer vision, YOLO-based object detection, precision agriculture, agricultural image processing, and smart agriculture. Tekirdag Namik Kemal University, Vocational School of Technical Sciences, Department of Computer Technology, Computer Programming Programme, Tekirdag, Turkey. Tel: +90-506 718 88 20, Email: ekahya@nku.edu.tr.

apple detection method, and this model has various configurations for fast and accurate detection of apples in complex natural environments[6]. They developed an optimized model called SE-YOLOv5x, which combines transfer learning and a visual attention mechanism. This model aims to discriminate and quickly locate grasses and lettuce plants in the field. Tests show that this model outperforms other existing methods with high accuracy rates[7]. A fast and accurate inspection model was developed with deep learning technologies for the automatic collection of blueberry berries. This model aims to collect blueberries with densely adhered berries and a structure with severe covering problems. The proposed blueberry recognition model integrates the NCBAM attention module to enhance feature extraction and a small-target detection layer to improve multi-scale recognition, while the C3Ghost module reduces model complexity[8]. A YOLOv5s-based method was developed for tree trunk and obstacle detection in an apple orchard. In this study, various improvements were made to increase the detection accuracy and speed of the model. The proposed model achieved MAP values ranging from 95.61% to 98.37% for images acquired over four seasons and met the requirement of real-time detection with high accuracy[9]. An automatic method for size classification of antler fungi using YOLOv5 and PSPNet was developed. This system provides real-time object detection and image segmentation to overcome the low efficiency and reliability of traditional manual classification[10]. A robot prototype designed to pick watermelons in greenhouse environments was also included in the studies. The developed robot detects watermelons with 89.8% accuracy with YOLOv5, a deep learning method, using a special end element to hold the fruits and cut the stems, and reaches a success rate of 93.3% in laboratory tests[11]. In addition, the YOLOv5-Litchi model was developed for litchi detection in complex natural environments by integrating CBAM, a small-object detection layer, Mosaic-9 augmentation, CIoU loss, and weighted-boxes fusion, improving mAP and recall by 12.9% and 15.0%, respectively[12]. Another study developed a system that analyzes the health status of trees using the deep learning method SSD (Single Shot MultiBox Detector). The developed system is capable of detecting different diseases with 92.5% accuracy by using the leaves and trunks of trees[13]. Finally, a system was developed using the YOLOv8+ model to detect and classify ripe strawberries. This system quickly and accurately determines the ripeness level of strawberries with 97.81% accuracy and provides the necessary data for selective harvesting[14]. In addition, an improved YOLOv9-based algorithm for detecting ripe tomatoes is presented. This model shows high performance in the detection of ripe tomatoes with 97.2% accuracy and 92.3% recall rates and provides a reliable solution in the picking process[15]. The kiwi is a fruit that attracts attention with its nutritional values and taste. This study begins with an Android application called KiwiDetector, which aims to quickly and accurately detect kiwis in orchards. Using MobileNetV2 and InceptionV3 as models, the studies provide significant time gains in the detection processes. They achieved 90.8%, 89.7%, 87.6%, and 72.8% correct detection rate (TDR) for MobileNetV2, quantized MobileNetV2, InceptionV3, and quantized InceptionV3, respectively[16]. Lim et al.[17] aimed to develop an autonomous pollination robot system that uses state-of-the-art object detectors such as Faster R-CNN and Single Stage Detector (SSD) to detect kiwifruit flowers. The proposed system showed results for accuracy, recall, and F1 score of 0.919%, 0.874%, and 0.889%, respectively, on the real-world dataset. Ulutas et al.[18] compared the performance of two deep learning methods, Faster R-CNN and Mask R-CNN, for automatic detection of kiwifruit in orchards. In the experiments, the ResNet50-based Mask R-CNN method was used. With various pre-trained architectures such as SqueezeNet and MobileNetV3, a mean accuracy (mAP) of 87.4% and 88.8% was achieved, respectively. Pawar et al.[19] emphasizes that timely detection of fruit diseases in India enables cost savings and compares the performance of hybrid machine learning and deep learning techniques. Experiments using hybrid machine learning and deep learning techniques have shown that the proposed hybrid CNN model effectively detects fruit diseases with a high accuracy of 97.10%. Xia et al.[20] developed a YOLOv7-based model for automatic detection of kiwifruit in New Zealand. The labeled dataset created from images of natural kiwi orchards was enriched with data augmentation techniques. Dewei et al.[21] extracted visual features from digital images using two deep learning-based models, YOLOv8 and CenterNet, to classify fruits as ripe and overripe. In addition, Selvakumari et al.[22] used a convolutional neural network (CNN) to classify and recognize fruit images and performed classification using support vector machines (SVM). Ma et al.[23] proposed a GG-CNN2-based approach to estimate the handle angle in kiwi harvesting to improve the efficiency of handle recognition. The ability of GG-CNN2 to provide optimal handle configuration along with depth and color images helps increase the harvesting rate by reducing interactions with neighboring fruits. Banerjee et al.[24] successfully detected five different diseases with an accuracy of 83.34% by using machine learning methods (CNN and SVM) for automatic detection of leaf diseases in kiwifruit. Ebrahimi et al.[25] aimed at automatic and non-destructive detection of bruise damage on kiwifruit in three classes using localized spatial-spectral near-infrared (NIR) hyperspectral images. For 2D-CNN-PreActResNet and 2D-CNN-GoogLeNet, the CCRs for immature fruit were 96% and 95%, respectively, while the CCRs for mature fruit were 91% and 98%, respectively.

Future research will be mindful of scalability and flexibility so that ongoing growth of the applicability of object detection models to agriculture can be achieved. Developing models that are able to operate well across various environments, be easily integrated into robotic platforms, and provide real-time decision-making will be critical to furthering agricultural automation[26]. Furthermore, the development of standardized measurement instruments for assessment specially designed for use in agriculture will make it possible to more closely measure model performance and guide future studies. The progress of object detection technologies, particularly with the YOLO series, has ensured a sound platform for integrating automatic systems in modern agriculture[27].

The new member of the family, YOLO11, is a big step forward in that it offers a more advanced architecture for greater accuracy, faster inference, and overall improved performance for low-resource situations. From this generation onwards, the YOLO model clearly stands out in that it includes a massive architecture, and with it, the ability to learn complex patterns, including being able to distinguish between kiwis and their environment under dense foliage or against harsh light[28]. However, to exploit the potential of such models, appropriate hyperparameter tuning is required to calibrate the performance of the model to the needs of a specific application. Furthermore, data augmentation techniques, such as rotation, luminance, and Gaussian distribution noise addition, significantly increase the size of the training sets, making the model robust and more general. Research is particularly intensifying in response to the growing worldwide interest in precision agriculture and sustainable agriculture. Research focuses on harvesting systems that mechanize agricultural tasks through machine learning algorithms to solve labor crises and minimize the environmental impact of agricultural operations[29]. By reducing human involvement, these harvesting systems can prevent fruit damage while maintaining consistent product quality. Furthermore, by selecting ripe fruits with high accuracy, they can facilitate timely harvesting and minimize post-harvest loss[2]. The adoption of such technologies in agricultural practices is part of larger initiatives to improve agricultural operations and sustain productivity.

However, to get the best out of such models, an optimal optimization process must be applied. Some lightweight versions of the YOLO models, such as YOLO11n and YOLO11s, have been optimized to overcome these limitations. Even though such variants provide improved inference efficiency and lower hardware demands, they sacrifice some precision, and further research is required to balance performance and efficiency.

The YOLO11x-mod in the study uses new hyperparameter settings instead of the default hyperparameter settings. Developments in object sensing models, especially in the YOLO series, have comprehensively supported the potential for automation in agriculture. The inherent heterogeneity of garden conditions, the integration of detection models with robotic systems, and their adaptability to various garden structures and conditions are still to be explored. There is also no traditional evaluation metric other than the traditional sensitivity, recall, and mAP50 measurements, and it is difficult to compare the overall performance of the models in agricultural tasks. Despite these advances, a significant research gap is the lack of use of deep object detection models for individual crop harvesting, such as kiwifruit. The inherent variability of kiwifruit orchards and the sensitivity of fragile structures need specially tuned models to address these specific issues[30]. The customized development of YOLO11x-mod carried out in the study enables the model to be optimized for specific kiwi detection tasks. Therefore, YOLO11x-mod is more sensitive, recalls with fewer false positives and missed detections, and is computationally lighter. Despite all these improvements, object detection models in agriculture are not easy to use. Orchards are highly variable, with changing light, leaf density, and fruit sizes[31]. All these parameters can affect detection accuracy by causing false positives or missed detections. To overcome these challenges, researchers have tried to incorporate contextual information such as texture, color, and spatial relationships into object detection models[32]. This would help models to better discriminate between target objects and the surrounding background, especially when objects such as kiwifruit are hidden under leaves or branches. The significant proportion of leaf cover and intrinsic heterogeneity of kiwifruit fields may limit the ability of models to distinguish kiwifruit fruits from the background and therefore lead to false detections or false positives. The innovations offered by object detection technologies in agriculture are accelerating the transformation of traditional methods, particularly in fruit harvesting. Automated systems offer significant benefits in agriculture, providing ever-increasing efficiency, reducing the need for human labor, and improving product quality. However, the challenges faced by existing models, in particular the variability of the natural environment and the low diversity of datasets, limit the effectiveness of these technologies.

In this context, the actual contribution of the current study is not only to improve performance by tuning the hyperparameters. It is also about developing a systematic learning strategy aimed at overcoming the fundamental limitations resulting from agricultural conditions. These strategies were not randomly selected but specifically designed to address the fundamental challenges in agricultural scenes, namely the detection of small objects, the detection of fruits hidden under leaves, and variable lighting conditions. The proposed model YOLO11x-mod achieves higher accuracy than previous models in small object detection, fruit hidden under dense foliage detection, and variable lighting conditions by using AdamW optimization, low weight drop, reasonable learning rate, and a combination of dropout and mosaic. In addition, the predictive performance of the model was evaluated not only using classic metrics (precision, recall, F1-score, AP at IoU=0.50 (mAP@50), and mean AP over IoU thresholds of 0.50-0.95 (mAP@50-95)), but also using error metrics (MAE, RMSE, MAPE, MSE). The results were therefore statistically validated. This comprehensive analysis shows that our work goes beyond similar studies in the literature by demonstrating not only an increase in accuracy, but also in generalization ability and error distribution. The ablation analysis clearly demonstrated the contribution of each individual change, thus defining the methodological innovation not only by tuning the parameters but also by applying these settings holistically to solve the critical challenges in agricultural imagery. In this respect, the study differs from previous YOLO-based applications in the literature and makes a significant contribution to the field of precision agriculture, both technically and practically.

2 Materials and methods

2.1 Materials

2.1.1 Data collection and augmentation

This study consists of two main parts: Training and test data. It comes from the kiwi-workspace/kiwi-v1.0 dataset that I created from kiwi photos published on the Roboflow Universe platform (https://universe.roboflow.com/kiwi-workspace/kiwi-v1.0). The dataset contains kiwifruit images taken under different environmental conditions. The original dataset contains a total of 420 images, which were taken under different lighting conditions (sunny, cloudy and shady) so that the effect of different environmental conditions on model performance can be analyzed. The images have a resolution of 1024×1024 pixels and were scaled to 640×640 for model training with corresponding to YOLO family.

The labeling was performed on the Roboflow platform, where each kiwifruit was marked with a rectangular bounding box. Roboflow's automatic labeling tools were used for this process, followed by manual checks to ensure accuracy. A total of 22 750 kiwi objects were labeled. For data augmentation, Roboflow's augmentation tools were used, and processes such as reflectance, brightness change, contrast adjustment, and the addition of Gaussian noise were applied to the images. In addition, the mosaic augmentation technique, which is important for YOLO models, was applied. As a result of these processes, the dataset was expanded to 928 images. The entire training and testing process was performed on the Roboflow platform, with a ratio of 7:2:1 between training, validation, and testing, respectively. This study was not conducted to develop a new dataset, but to fairly compare existing YOLO family models under the same conditions and determine the most appropriate hyperparameter strategy. Therefore, the details of the dataset were extensively expanded in the revision, and the source used, the labeling method, and the augmentation processes were clearly explained.

This comprehensive data preparation process provides a more realistic scenario that includes different lighting conditions, different fruiting positions, and overlapping situations, unlike previous studies that only focused on limited conditions. This has improved the generalizability of the developed YOLO11x-mod model in agricultural applications. Examples of kiwi images used in the training group are shown in Figure 1.

Figure 1 Examples of training set images

To improve the image features, prevent overfitting, and increase the robustness of the model, the Augmentor tool is used in this study for data augmentation prior to network training. Augmentor is a Python library developed for image augmentation operations in computer vision projects. Kiwifruit grown together can experience light shadows, leaf shadows, and skewed fruit growth during detection. Augmentor provides a set of user-friendly methods and tools to perform various augmentation operations on images. In this study, the dataset is expanded by vertical flipping, brighter, darker, golden, and augmentation procedures. This data augmentation procedure increases the size of the dataset to 928 images. 22 750 kiwis are labeled. Figure 2 shows an example of images with data augmentation next to the original image after data augmentation.

a. Original figure

b. Vertical flip c. Brighter d. Darker e. Golden

Figure 2 Data magnification processing

In the study, the mosaic data augmentation method for YOLOv8 was used first. Mosaic data augmentation increases the sample size by using four images simultaneously during training. This method increases the number of small targets and improves the model's detection capabilities by randomly reducing large images to smaller sizes. Mosaic data augmentation creates more training patterns by merging images; this increases the generalization ability of the model and helps it to better understand the relationships between different objects[15]. Mosaic data augmentation improves target detection performance by adapting to the size, position, and background of the target. Mosaic scales the images before merging so that small targets do not lose their important features. After merging, the bounding box coordinates are adjusted to ensure the accuracy of the labeling information. By combining conventional data augmentation techniques with the mosaic algorithm, the training set is expanded to 928 images, and a comprehensive kiwi dataset is created. The entire dataset is divided into training, validation, and test sets. This ratio is 7:2:1. Figure 3 shows labeling examples used for the mosaic data expansion.

Figure 3 Examples of labeling used for mosaic data augmentation

2.1.2 You Only Look Once (YOLO) models

YOLO is a deep learning object detection model that operates in real time and is highly efficient and fast. Compared to previous versions, YOLOv8 is faster and more efficient in object detection. This means that the model consumes less computational power, which makes it perfect for use in mobile and embedded systems.

The model boasts several improvements to provide more precision. These have been achieved by applying larger datasets and more complex architectural designs. YOLOv8 has applications in a variety of applications for real-time object detection. YOLOv8 utilizes a Convolutional Neural Network (CNN) architecture akin to previous iterations but has deeper, more complex layers. The model is then trained using extensive and diverse datasets, significantly boosting its ability to detect many forms of objects. YOLOv8 returns object detection output as boxes and labels, facilitating ease of comprehension by the user in realizing what results are being attained.

YOLOv10 and YOLO11 are the latest members of the You Only Look Once (YOLO) series, the state-of-the-art for real-time object detection. Developed by researchers at Tsinghua University, they were released on 23 May 2024 with significant improvements over previous versions. YOLOv10 and YOLO11 are characterized by faster inference times with fewer parameters, and strive to meet the increasing demands of various computer vision applications such as surveillance, autonomous driving, and robotics. Performance comparisons reveal that YOLOv10 is 1.8 times faster compared to the RT-DETR-R18 model and has a similar average precision (AP) on the COCO dataset. It is also noted that this model offers efficiency gains with 2.8 times fewer parameters and Floating Point Operations Per Second (FLOPs). In particular, the YOLOv10-B version performs similarly, with 46% lower latency and 25% fewer parameters than YOLOv9-C.

The architecture of YOLO11 includes several important improvements to enhance performance. It incorporates advanced features derived from the success of previous YOLO releases and further reduces latency and parameter count. "With YOLO11, we aimed to create a model that combines power and practicality for real-world applications," said Glenn Jocher, CEO and founder of Ultralytics, emphasizing the design intent of the model. The optimized architecture not only supports real-time detection but also responds to specific challenges in different industries, expanding the potential use cases of the model. Early evaluations indicate that YOLO11 reduces computational complexity and latency while maintaining competitive detection performance. Similar to the advances in YOLOv10, it achieves significant delay and parameter reductions while maintaining performance levels similar to RT-DETR models. Figure 4 shows the block diagram of YOLO11.

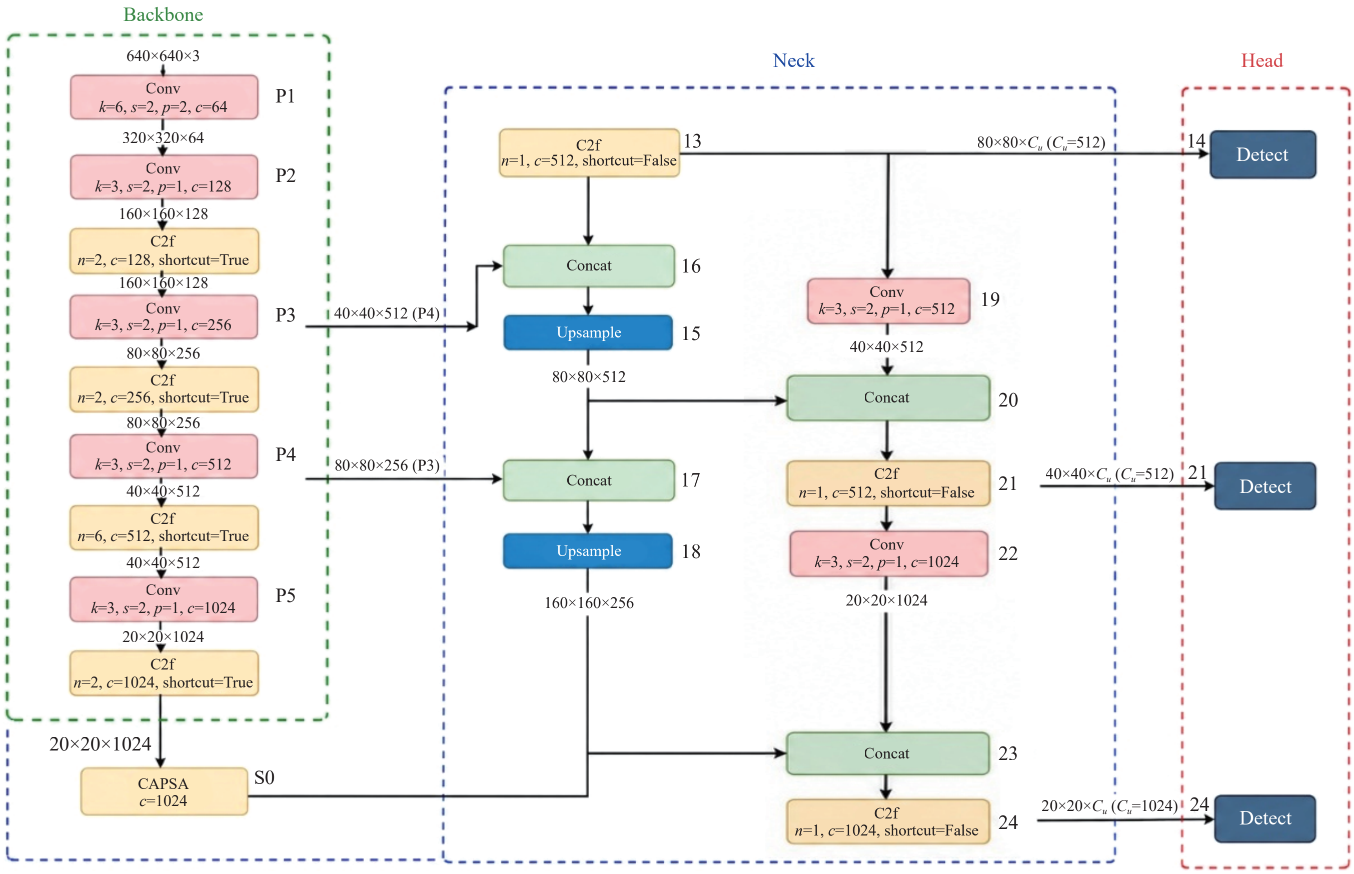


Figure 4 YOLO11 structure diagram

2.2 Methods

2.2.1 Evaluation criteria

To test the effectiveness of the proposed method, different evaluation criteria were used for three deep learning algorithm models. First, recall, precision, F1 score, and mean average precision (mAP) were used as evaluation criteria to evaluate the model for the one-step YOLO object detection algorithm, which includes the following four aspects.

The equations used in the calculation are shown below.

$$P=\frac{\mathrm{TP}}{\mathrm{TP{+}FP}} \tag{1}$$

$$R=\frac{\mathrm{TP}}{\mathrm{TP{+}FN}} \tag{2}$$

$$\mathrm{F1}=\frac{P\mathrm{x}R\mathrm{x}2}{P+R} \tag{3}$$

where, TP is the correctly predicted positive samples as positive, FP is the negative samples falsely predicted as positive, FN is the positive samples falsely estimated as negative, and TN is the correct prediction of negative samples as negative.

Average mean precision (mAP): The average mean precision is a summary metric that reflects precision at various recall levels. It is calculated by averaging the average precision (AP) values across all classes.

$$\mathrm{AP}=\int_0^1 P(R) \tag{4}$$

$$\mathrm{mAP}=\frac{1}{N}\sum_{i=1}^{n}\mathrm{AP}_i \tag{5}$$

$$\mathrm{mAP@50\%}=\frac{1}{N}\sum_{i=1}^{n}\mathrm{AP@0.5} \tag{6}$$

where, AP_i is the average precision for class i. N is the number of classes.

Accuracy (precision): The proportion of true positive instances among all instances predicted as positive.

Recall: The proportion of all true positive samples that are correctly predicted as positive.

2.2.2 Error metric evaluation

The mean squared error (MSE) is a statistical evaluation tool that measures how far a model's predictions are from the actual values. The MSE is often used in regression analysis and plays an important role in evaluating the success of the model.

$$\mathrm{MSE}=\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2 \tag{7}$$

where, n is the total number of data points, y_i is the true value, and $\hat{y}_i$ is the value predicted by the model.

The mean absolute error (MAE) is another statistical evaluation tool that measures how close a model's predictions are to the true values. As the MAE is calculated using the absolute values of the error values, large errors are weighted less heavily, and it is considered a more robust measure of error.

$$\mathrm{MEA}=\frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i| \tag{8}$$

where, n is the total number of data points, y_i is the true value, and $\hat{y}_i$ is the value predicted by the model.

Mean absolute percentage error (MAPE) is a performance evaluation tool that measures the percentage deviation of a model's predictions from the true values. MAPE makes the model's performance easier to understand by taking into account both the size of the error and the percentage of this error in relation to the true values.

$$MPEA = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i-\hat{y}_i}{y_i}\right|x100 \tag{9}$$

The root mean square error (RMSE) is a statistical measure that indicates how far the predicted values of a model are from the true values, which is often used to understand the overall performance of the model.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2} \tag{10}$$

2.2.3 Experimental setup and hyperparameter settings

All experiments in this study were performed with an NVIDIA A100 GPU (40 GB VRAM) in the Google Colab Pro+ environment. This powerful computing infrastructure ensured that all YOLO models were trained under equal, fair, and comparable conditions. The YOLO deep learning model and its sub-models (YOLOv8, YOLOv10, and YOLO11) were trained separately. Early stopping for the validation loss was set to 30 epochs, and the total training time was set to 100 epochs. The size of the input image was set to 640×640 pixels.

The changes made to the YOLO11x architecture applied various hyperparameters and optimization techniques to improve performance. To maintain balance during the training process, the epoch number (d) was set to 30 to prevent overfitting and ensure sufficient learning. The learning rate (e) was set to 0.005; this value ensures balanced learning by keeping the update rate of the model's weights at a moderate level. A weight drop (w) of 0.0001 was added for regularization, which prevents overfitting and increases the generalization capability. AdamW was chosen as the optimization algorithm; this algorithm retains the advantages of classic Adam, but applies the weight drop more effectively. In the data augmentation, the mosaic technique was used to combine different images, increasing the diversity of data and improving the model's success in recognizing small objects. This configuration allowed the model to better adapt to variations in agricultural scenes. The balance between the learning rate and the weight drop ensured a more stable learning of the model, while the use of AdamW made the optimization process more consistent and efficient.

Not only are accuracy metrics (precision, recall, F1-score, AP at IoU=0.50 (mAP@50), and mean AP over IoU thresholds of 0.50–0.95 (mAP@50-95)) reported in the experimental results, but also indicators of computational efficiency such as FPS, number of parameters, and GFLOPs values. These additional metrics make it possible to evaluate models not only in terms of accuracy, but also in terms of computational cost and speed efficiency. Thus, the applicability in embedded systems and real-time applications was discussed on a theoretical level. Overall, these changes have improved the accuracy, generalization capability, and success of the YOLO model in detecting small objects. The modified structure is shown in Figure 5.

Hyperparameter values used in the study are listed in Table 1 for YOLOv8, YOLOv10, YOLO11, and YOLO11x-mod, respectively.

Table 1 lists various characteristics of the different YOLO models. As far as the learning rate (lr0) is concerned, YOLOv8 has the highest speed with a learning rate of 0.01. The YOLOv10 and YOLO11 models follow a more cautious learning process with a learning rate of 0.001. It goes without saying that the YOLO11x-mod model is balanced with a learning rate of 0.005.

Table 1 Hyperparameter values of YOLOv8, YOLOv10, YOLO11, and YOLO11x-mod

Parameter	Model values			
	YOLOv8 (default)	YOLOv10 (default)	YOLO11 (default)	YOLO11x-mod
Learning rate	0.01	0.001	0.001	0.005
Momentum	0.9	0.937	0.9	0.937
Weight_decay	0.0005	0.0005	0.0005	0.0001
Optimizer	Adam	Adam	Adam	AdamW
Shear	0.3	0.3	0.3	0
Perspective	0.001	0.001	0.001	0
Mosaic	True	True	True	0.8
Mixup	True	False	True	0.2
Dropout	0.5	0.5	0.5	0,4
Device	CUDA	CUDA	CUDA	0
Amp	True	True	True	False
Batch:n,s,m,l	16,32,64,128	16,32,64,128	16,32,64,128	32
Batch:x	32	32	32	16
Rotate	True	True	True	0.5
Rotate_limit	15	15	15	−10,1
Clahe	True	True	True	0.1
Clip_limit	2.0	2.0	2.0	4.0
Title_grid_size	8	8	8	8,8
Random_scale	True	True	True	0.5
Blur	True	True	True	7
Blur_limit	7	7	7	3

When analyzing the parameter for the weight drop, it can be seen that YOLO11x-mod has a lower weight drop with a value of 0.0001. This lower weight drop helps the model to be more flexible and to generalize better.

As for the optimization methods, all models preferred the Adam algorithm; YOLO11x-mod preferred the AdamW algorithm. The reason for using the AdamW algorithm is its potential to increase the overall performance by including the weight decay in the optimization process.

1) Calculations

First Moment Estimate (m_t):

$$m_t = \beta_1 \cdot m_{t-1} + (1-\beta_1) \cdot g_t \tag{11}$$

where, m_t is the current value of the first moment (the running average of gradients); β_1 is the exponential decay rate for the first moment estimate (usually set to around 0.9); m_{t-1} is the previous first moment value; g_t is the gradient of the loss function with respect to the parameters at time step t.

Second Moment Estimate (v_t):

$$v_t = \beta_2 \cdot v_{t-1} + (1-\beta_2) \cdot g_t^2 \tag{12}$$

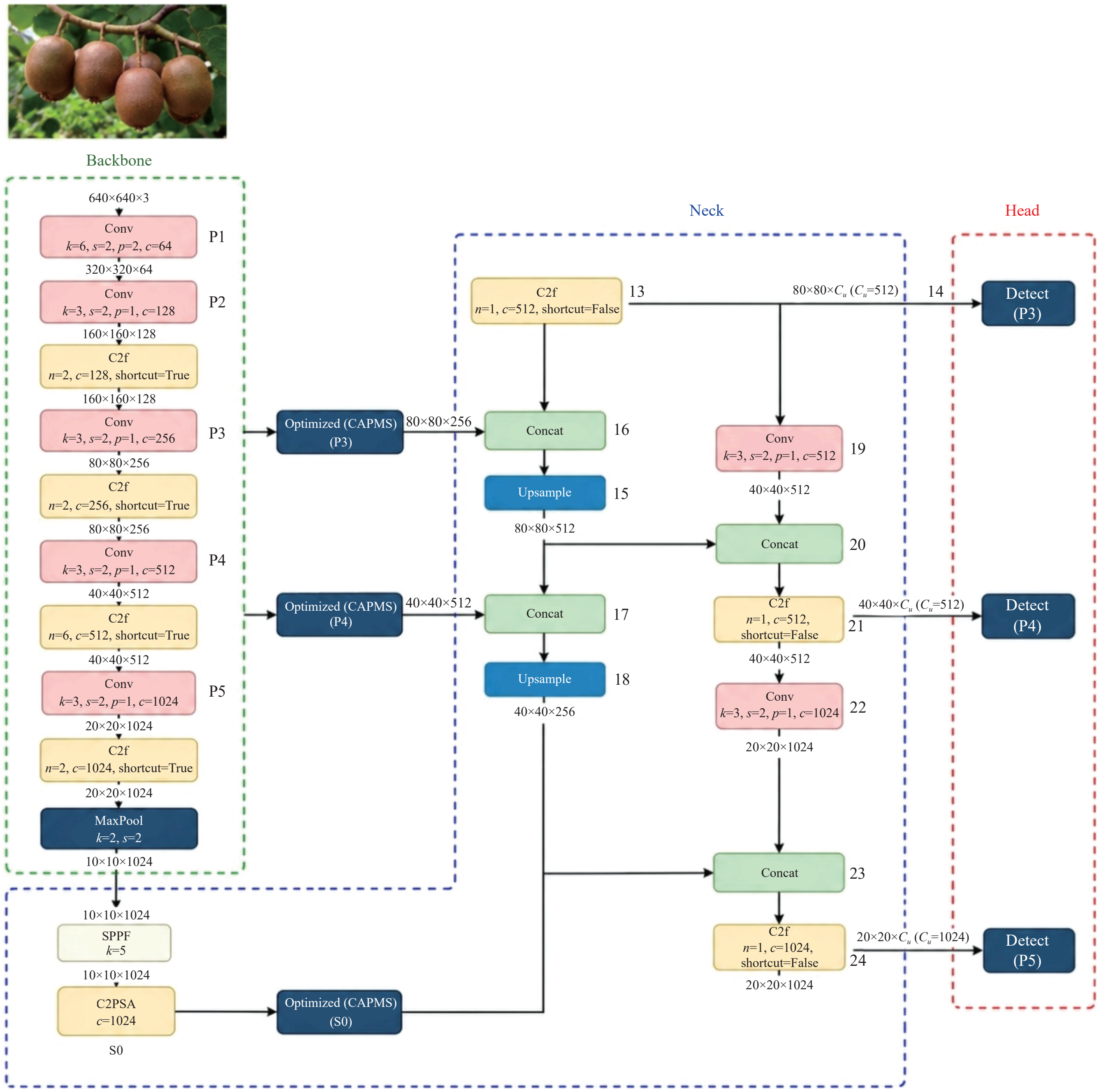
where, v_t is the current value of the second moment (the running average of the squared gradients); β_2 is the exponential decay rate for the second moment estimate (usually set to around 0.999); v_{t-1} is the previous second moment value; g_t^2 is the square of the gradient of the loss function at time step t.

Bias-Corrected First Moment ($\hat{m}_t$):

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{13}$$

where, $\hat{m}_t$ is the bias-corrected estimate of the first moment, which compensates for the initialization bias toward zero in the first few iterations.

Bias-Corrected Second Moment ($\hat{v}_t$):



Note: *d (number of epochs) value 30, e (learning rate) value 0.005, w (weight_decay) value 0.0001.

Figure 5 YOLO11x-mod structure diagram

$$\hat{v}_t = \frac{v_t}{(1-\beta_2^t)} \tag{14}$$

where, $\hat{v}_t$ is the bias-corrected estimate of the second moment, also correcting the bias from initial values; β_2 is the hyperparameter controlling the decay rates for the moments.

2) Parameter Update (AdamW)

$$\theta_{t+1} = \theta_t - \eta\left(\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}\right) + \lambda \cdot \theta_t \tag{15}$$

where, θ_{t+1} is the updated parameter values after the update step; θ_t is the current parameter values before the update step; η is the learning rate, which controls how much to change the parameters in the update; ε is a small constant added for numerical stability to avoid division by zero (commonly set to 1×10^{-8}); and λ is a weight reduction coefficient that is applied directly to the parameter update.

The data expansion settings were set to "true" for all models. It contributes to the diversity of the model and to increasing the generalization capability. In YOLO11x-mod, the mosaic and mixing parameters were assigned values such as 0.8 and 0.2. The data augmentation strategy was designed to be simpler than in other models. As far as the number of epochs is concerned, the training time of all models is set to 100 epochs. To avoid excessive training and over-learning of models, the epoch value is set to 100.

As for the device and AMP (Automatic Hybrid Acceleration) features, all models offer GPU acceleration with CUDA support. However, the AMP value for YOLO11x-mod is set to "false", which can cause the model to run slower and consume more memory.

Finally, the image size and stack size were evaluated. While the image size (imgsz) remains constant for all models, the parameters batch:n,s,m,l vary. In particular, YOLO11x-mod uses a smaller batch size, while other models offer flexible batch sizes. This difference can lead to changes in memory utilization and learning speed during model training.

Google Colab (Colaboratory) is a free cloud-based Python coding environment provided by Google. With the Colab platform, users such as researchers, data scientists, and software developers can run and save Python code in the browser. Google Colab combines the powerful features of Jupyter notebooks with easy access to GPU (Graphics Processing Unit) and TPU (Tensor Processing Unit) computing resources by utilizing Google Cloud infrastructure. The dataset was analyzed using the Ultralytics software development framework.

3 Results

3.1 Comparison of ablation experiment results

The ablation analysis performed showed that the YOLO11 family offers significant improvements in terms of accuracy and stability compared to previous generations. The comparative performance results of the YOLOv8, YOLOv10, and YOLO11 models are listed in Table 2.

Table 2 Performance comparison of YOLOv8, YOLOv10, YOLO11, and modified YOLO11 models in terms of accuracy and computational efficiency

Model	Precision	Recall	F1 Score	mAP@50	mAP@50:95	FPS	Params (M)	GFLOPs	Memory/MB
YOLOv8s	0.8027	0.8158	0.8092	0.8477	0.655	115	11.2	28.6	20
YOLOv8n	0.8114	0.7949	0.8031	0.8416	0.639	160	3.2	8.7	13
YOLOv8m	0.7864	0.8218	0.8037	0.8447	0.660	85	25.9	78.9	42
YOLOv8l	0.7913	0.8342	0.8122	0.8511	0.664	58	43.7	165.2	78
YOLOv8x	0.7964	0.8192	0.8076	0.8434	0.658	41	68.2	258.5	110
YOLOv10s	0.8027	0.8087	0.8057	0.8534	0.650	108	11.5	30.0	22
YOLOv10n	0.7822	0.7834	0.7828	0.8353	0.633	155	3.5	9.0	13
YOLOv10m	0.8010	0.8181	0.8095	0.8552	0.659	78	26.5	82.0	45
YOLOv10l	0.8046	0.8074	0.8060	0.8559	0.657	51	46.0	170.0	80
YOLOv10x	0.8073	0.8160	0.8116	0.8538	0.658	38	70.0	265.0	115
YOLO11s	0.8021	0.8119	0.8070	0.8567	0.657	111	11.8	29.5	21
YOLO11n	0.8053	0.7995	0.8024	0.8497	0.638	158	3.5	9.0	13
YOLO11m	0.8068	0.8286	0.8175	0.8646	0.663	82	27.0	83.5	47
YOLO11l	0.8038	0.8189	0.8113	0.8595	0.666	54	47.5	172.0	82
YOLO11x	0.7993	0.8310	0.8148	0.8586	0.665	40	72.0	270.0	118
YOLO11s-mod	0.8257	0.8203	0.8069	0.8549	0.656	112	12.0	30.5	22
YOLO11n-mod	0.8346	0.8074	0.8002	0.8510	0.638	160	3.6	9.2	13
YOLO11m-mod	0.8221	0.8314	0.8081	0.8593	0.664	83	27.2	84.0	47
YOLO11l-mod	0.8182	0.8264	0.8035	0.8609	0.667	55	48.0	173.0	83
YOLO11x-mod	0.8176	0.8397	0.8068	0.8619	0.668	42	72.5	272.0	119

According to the results obtained, the YOLO11m model achieved the highest overall accuracy with 86.46% mAP@50 and 65.9% mAP@50:95, while the YOLO11x-mod model performed better, especially for small objects, with values of 86.19% mAP@50 and 66.8% mAP@50:95. When analyzing the F1 values, YOLO11m (81.75%) and YOLO11x (81.48%) stood out; YOLO11n-mod (83.46%) achieved the highest results for precision, and YOLO11x-mod (83.97%) for recall. In the speed analysis, the YOLOv8n (160 FPS) and YOLO11n-mod (160 FPS) models stood out with their real-time performance, while the YOLOv10x (38 FPS) and YOLO11x (40 FPS) models were limited due to their low speed. When evaluating parameter size, GFLOPs, and memory consumption, Nano models (~3.5 million parameters, ~9 GFLOPs, ~13 MB) are quite favorable in terms of hardware, while Xlarge models (~72 million parameters, ~270 GFLOPs, ~118 MB) offer high accuracy but are challenging for use in real-time systems. These results show that the YOLO11m and YOLO11x-mod models are the best options in terms of accuracy, while the YOLO11s-mod (mAP@50 85.49, 112 FPS) and YOLOv10s (mAP@50 85.34%, 108 FPS) models can be considered the most suitable candidates for practical applications in terms of the balance between speed and accuracy.

The main reason why the YOLO11x-mod model delivers the best results is the deeper layer structure and the greater parameter capacity achieved through changes to the architecture. This allows the model to gain an advantage, especially in learning complex object features and achieving high discriminative power in recognizing small objects. With results of 86.19% mAP@50, 66.8% mAP@50:95, and 83.97% recall, it outperforms all other models on accuracy metrics, simultaneously increasing precision and generalization success. Although the high number of parameters (~72.5M) and computational cost cause some speed penalty, the ablation study results in Table 3 prove that YOLO11x-mod is the most reliable and best-performing model, especially in academic and industrial applications that require high accuracy.

Comparisons were made between models of the same scale. The comparison between base and mod is listed in Table 4.

The ablation study showed the performance improvements of the individual mod versions of the YOLO11 family compared to the base model. All mod models showed increases in precision (+1.4%-2.9%) and recall (+0.7%-0.9%), while mAP@50:95 showed small but consistent gains (+0.1%-0.3%). These results indicate that each of the hyperparameter modifications applied made a positive contribution. In particular, the precision improvement achieved with AdamW and the recall increase achieved by the dropout-mosaic combination are directly reflected in the overall success of YOLO11x-mod. Thus, the performance improvement results not only from the model scaling, but also from the proposed hyperparameter strategies.

3.2 Comparative experiments of different models

The labeled images were tested with YOLOv8, YOLOv10, and YOLO11. In the study, YOLOv8, YOLOv10, and YOLO11 were tested with the entire trained set. The results were analyzed, and the three best models were found. These are YOLOv8l, YOLOv10l, and YOLO11m, respectively. The three best models were again analyzed against each other. The YOLO11m model from the YOLO11 family was found to be the best model. The hyperparameters of YOLO11 were therefore changed manually. The modified YOLO11 models were again analyzed against each other. As a result of the analysis, the best model, YOLO11x-mod, was determined. The YOLO11x-mod value obtained was determined again and compared with the three best models, YOLOv8l, YOLOv10l, and YOLO11m. The results of the comparison are listed in Table 2. These models were evaluated, and the best submodels were selected. YOLO11 was determined to be the best model. YOLO11 was modified and compared with the best models found. Table 5 lists the comparison results and presented values. Figure 6 shows a general comparison of YOLO versions.

Table 3 Performance comparison of selected YOLO models and the proposed YOLO11x-mod

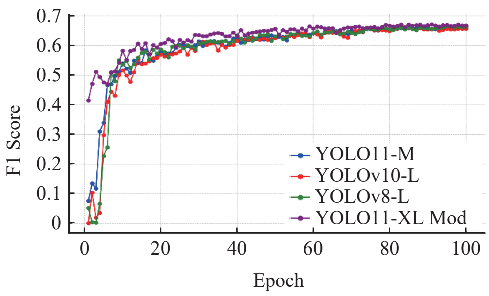
State	Model	Precision	Recall	F1	mAP@50	mAP@50:95	Comment
1	YOLOv8 family (n, m, l, x, s)	The best: 0.7913	The best: 0.8342	The best: 0.8122	**0.8511 (YOLOv8l)**	~0.664	The best performance in the YOLOv8 family was observed in YOLOv8l.
2	YOLOv10 family (b, l, m, n, s, x)	The best: 0.8046	0.8074	0.8060	**0.8559 (YOLOv10l)**	~0.657	In the YOLOv10 family, the highest mAP50 was obtained in YOLOv10l.
3	YOLO11 family (n, s, l, x, m)	0.8068	0.8286	**0.8175**	**0.8646 (YOLO11m)**	~0.684	The best model in the YOLO11 family was the YOLO11m.
4	YOLO11x (Base)	0.7993	0.8310	0.8148	0.8586	~0.668	Base YOLO11x model.
5	YOLO11x-mod (Final)	**0.8176**	**0.8397**	0.8068	**0.8619**	~0.669	Modified YOLO11x; significant increase in precision, recall, and mAP50.

Table 4 Performance changes (Δ) between baseline and modified YOLO11 models

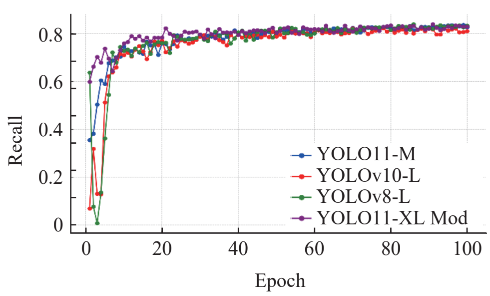
Model	PrecisionΔ	RecallΔ	F1Δ	mAP@50Δ	mAP@50:95Δ
YOLO11s- YOLO11s-mod	+0.0236	+0.0084	−0.0001	−0.0018	−0.001
YOLO11n- YOLO11n-mod	+0.0293	+0.0079	−0.0022	+0.0013	0.000
YOLO11m- YOLO11m-mod	+0.0153	+0.0028	−0.0094	−0.0053	+0.001
YOLO11l- YOLO11l-mod	+0.0144	+0.0075	−0.0078	+0.0014	+0.001
YOLO11x- YOLO11x-mod	+0.0183	+0.0087	−0.0080	+0.0033	+0.003

Table 5 Comparative performance of selected YOLO models and the proposed YOLO11x-mod based on key detection metrics

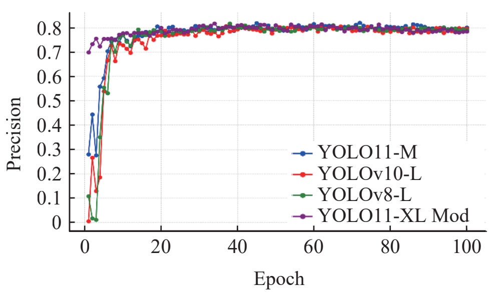
Model	Precision	Recall	mAP@50	F1 Score	mAP@50:95
YOLOv8l	0.7913	0.8342	0.8511	0.8122	0.6640
YOLOv10l	0.8046	0.8074	0.8559	0.8060	0.6570
YOLO11m	0.8068	0.8286	0.8646	0.8175	0.6630
YOLO11x-mod	0.8176	0.8397	0.8619	0.8068	0.6680



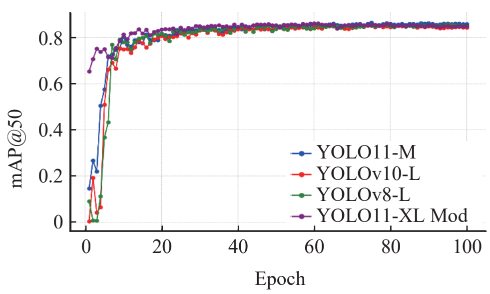
a. F1 score



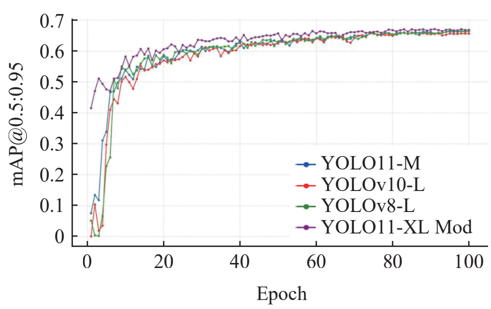
b. Recall



c. Precision



d. mAP@50



e. mAP@50:95

Figure 6 General comparison of YOLO versions F1 score, recall, precision, mAP@50, and mAP@50:95

According to the values in Figure 6 and Table 2, the general comparison of YOLO versions is as follows:

These values show that usable models can be created quite quickly with medium-sized GPUs or similar hardware. However, difficult conditions such as dense leaf cover during kiwifruit harvest or small fruits with different maturity levels may affect the detection performance. Although YOLOv8 and YOLOv10 provide a good basic solution, they fall behind the more advanced versions in environments with dense vegetation or in scenarios requiring high detection sensitivity, as shown by the data in the table. These values prove that they work quite efficiently, especially with complex kiwifruit images.

3.2.1 Comparisons between the best models of the versions

According to the values in Figure 6a and Table 2, the F1 score of the YOLOv8l model is determined as 0.8122. This value shows that the model effectively provides a balance between true positives and negatives. A high F1 score indicates that the model performs well in terms of both precision and sensitivity. The YOLOv10l model, on the other hand, performed lower with an F1 score of 0.8060. This indicates that the model may make errors in making positive predictions in some cases. The F1 score of the YOLO11x model is quite remarkable, at 0.8148. YOLO11x provides a better balance compared to other models by offering a good combination of both precision and sensitivity. The highest F1 score belongs to the YOLO11x-mod model, and this value was determined as 0.8068. The performance of this model reveals that it is quite effective in identifying true positives and avoiding false positives. According to the F1 score values, the YOLO11x model shows the highest success with a score of 0.8148 and therefore stands out as the most balanced model. The YOLOv8l model is right behind it with a score of 0.8122, while the performances of the other models are slightly behind. YOLOv10l, as the model with the lowest F1 score, shows that it may have certain difficulties in positive predictions in some cases. According to the values in Figure 6b and Table 2, the recall value of the YOLOv8l model was determined as 0.8342. This high value shows that the model is quite successful in recognizing positive examples. In particular, it exhibits effective performance in identifying true positives. The YOLOv10l model draws attention with its recall value of 0.8074. This value is slightly lower than the YOLOv8l model, thus indicating that this model is slightly less effective in recognizing positive examples. However, YOLOv10l still exhibits an acceptable performance. The recall value of the YOLO11x model is 0.8310. YOLO11x shows a performance very close to YOLOv8l and offers a high level of sensitivity. The highest recall value belongs to the YOLO11x-mod model with 0.8397. This model shows slightly superior performance compared to other models in correctly identifying true positives. This high value shows that the model pushes the limits of its error-free and complete detection capacity. All models have high recall values, and the YOLO11x-mod model shows better performance than the others. This shows that these models make fewer errors in identifying true positives. It should also be emphasized that even small differences between models can be important in certain applications. In particular, if error-free detection is a critical requirement, the YOLO11x-mod model is the best choice.

According to the values in Figure 6c and Table 1, the precision value of the YOLOv8l model was determined as 0.7913. This high value shows that the model generally produces correct results in its positive predictions and is effective in avoiding false positives. The YOLOv10l model has a precision value of 0.8046. This value represents a higher success rate than the YOLOv8l model and shows that the model makes fewer errors in its positive predictions. In particular, it exhibits good performance in terms of its ability to make correct classifications. The precision value of the YOLO11x model was determined as 0.7993. This result indicates that the model shows quite good success, but offers less precision compared to some other models. The highest precision value belongs to the YOLO11x-mod model, and this value was determined as 0.8176. It has a very high precision rate, showing that the model makes very few errors in positive classifications and provides reliable results. According to the values, the YOLO11x-mod model stands out as a more reliable option than other models with the highest precision value of 0.8176. This result reveals that the model offers high reliability in positive predictions and minimizes false classifications.

According to the values in Figure 6d and Table 2, the mAP@50 value of the YOLOv8l model is determined as 0.8511. This value shows that the model provides high success in object detection. The YOLOv10l model shows slightly higher performance in this criterion with a mAP@50 value of 0.8559. This model shows that it is particularly effective in terms of object detection accuracy. The mAP@50 value of the YOLO11x model is 0.8586, which reveals that it is the model with the best performance. YOLO11x can detect objects most accurately at IoU level 0.50. The highest mAP@50 value belongs to the YOLO11x-mod model with 0.8619. This model stands out in terms of overall object detection success and shows superior performance compared to other models. While all models show high performance in general, the YOLO11x-mod model offers significant superiority in object detection. This finding shows that the ability of the model in question to detect objects is more effective than other models.

According to the values in Figure 6e and Table 2, the mAP@50:95 value of the YOLOv8l model is determined as 0.664. This value represents the general success and consistency of the model in object detection over a wide range. However, this value remains lower than that of other models. The YOLOv10l model draws attention with its mAP@50:95 value of 0.657. This means that the model is less consistent and may encounter certain difficulties in the object detection task. Therefore, the overall performance of this model does not seem to be sufficient in wider IoU ranges. The mAP@50:95 value of the YOLO11x model reaches 0.665. This shows that the model provides slightly better performance in a wide object detection range. The highest mAP@50:95 value belongs to the YOLO11x-mod model, and this value is recorded as 0.668. This model provides noticeable success in detecting objects in a wide range and thus stands out as the model with the best performance. In general, the mAP@50:95 values are quite similar among all models. However, the YOLO11x-mod model gives the best results in this area with 0.668. This finding shows that the model is more effective than the other models in a wide object detection range.

3.2.2 Superiority of YOLO11x-mod for kiwifruit harvesting and final evaluation

The results obtained in kiwifruit harvesting applications show that the “mod” versions of the YOLO11 family, especially the “YOLO11x” variant, can successfully manage very difficult conditions. Different lighting conditions or the presence of kiwifruits under dense leaf cover are often factors that can cause errors in detection systems. However, these tables and experimental analysis show that YOLO11x-mod stands out in the following areas:

High precision (0.8176) and recall (0.8397): Reducing false positives (precision) and ensuring that fruits are not missed (recall) provide double benefits in kiwifruit harvesting. High values of these metrics reduce unnecessary processing times of robotic systems while maximizing harvesting efficiency.

mAP@50=0.8619: The model detects kiwifruits of different sizes, shapes, and partial obstacles with high average sensitivity. Even with kiwifruit varieties with different maturity levels or varieties, continuous success can be achieved.

F1=0.8068: A relatively high F1 score reflecting the balance of precision and recall shows that the model offers a versatile performance in general. Achieving such an F1 score is remarkable, especially when variables such as branches, leaves, and shadows in the field are taken into account.

Large parameter capacity of the "YOLO11x" variant: A large architecture can learn complex patterns (color tones, leaf-branch arrangements, field topography, etc.) in more detail. If sufficient hardware (GPU memory, processing power) is available, YOLO11x-mod has the potential to scan large-scale orchards with high speed and precision.

mAP@50:95=0.6680: The mAP@50:95 values of the four models are YOLOv8l (0.6640), YOLOv10l (0.6570), YOLO11m (0.6630), YOLO11x-mod (0.6680), respectively. According to these values, YOLO11x-mod is the best model compared to the other models, as it has the highest mAP@50:95 score (0.6680).

FPS=52.63: Considering the frames per second (FPS) and inference time data, YOLOv10l was found to be the fastest model. However, in terms of overall performance and accuracy criteria, the YOLO11x-mod model was found to be the best option. YOLO11x-mod is considered to offer superior performance in professional applications requiring high accuracy. Therefore, YOLO11x-mod is the best model compared to the others. Estimation result screenshots are shown in Figure 7.

Figure 7 Prediction screen YOLOv8l, YOLOv10l, YOLO11m, YOLO11x-mod

Figure 7 shows the best detection results of the four best models. As can be seen in Figure 7d, YOLO11x-mod shows its superiority over the other models in the prediction results.

3.3 Comparisons of error metric evaluation

The error evaluation of the obtained results was carried out with MSE (mean squared error), MAE (mean absolute error), MAPE (mean absolute percentage error), and RMSE (root mean square error). The results of the evaluation of the error metrics of the other three models according to the YOLO11x-mod model are shown in Table 6.

Table 6 Evaluation of the error metrics of the other three models according to the YOLO11x-mod model

Metric	MSE	MAE	MAPE/%	RMSE
Precision	0.000 398	0.0192	2.35%	0.019 96
Recall	0.000 383	0.0155	1.85%	0.019 57
F1 Score	0.000 031	0.0047	0.59%	0.005 59
mAP@50:95	0.000 048	0.0060	0.90%	0.006 98
mAP@50	0.000 054	0.0067	0.78%	0.007 38

Table 3 shows that the F1 score metrics have the lowest error rates. The difference is somewhat more pronounced for the precision and recall metrics. The errors are at a low level, which shows that the models make predictions that are quite close to the true value of 0.70. It can be seen that the YOLO11x-mod model shows a very consistent superiority in all metrics compared to the other models. This shows that the YOLO11x-mod model we developed has a high accuracy rate. It is superior to other existing models due to the efficiency improvements.

3.4 Bland-Altman analysis

The YOLO11m and YOLO11x-mod models were compared using the Bland-Altman method. The mean difference (bias) was small, and the 95% confidence limits were within a narrow range, suggesting that there is no systematic bias between the two models. However, YOLO11x-mod was found to score consistently higher. The plot of the Bland-Altman analysis is shown in Figure 8.

The Bland-Altman analysis illustrates the comparative performance of the YOLO11x-mod model against other strong models (YOLOv8l, YOLOv10l, and YOLO11m). The results show that YOLO11x-mod has positive bias values compared to YOLOv8l and YOLOv10l, indicating that it systematically achieves higher accuracy. Compared to YOLO11m, a small negative bias was observed, suggesting that YOLO11m offers a limited advantage in the overall mAP@50 score. However, when looking at the precision and recall metrics, YOLO11x-mod showed improvements of 2.3% and 1.0%, respectively. In addition, YOLO11x-mod achieved the highest score of around 0.79 when analyzing small objects (AP_small), which is a clear advantage over YOLO11m. The fact that YOLO11x-mod stands out with higher precision and recall values, especially when recognizing small objects, makes this difference practically significant for agricultural applications. These results confirm that the modified YOLO11x-mod is a strong alternative to the base and previous generation models.

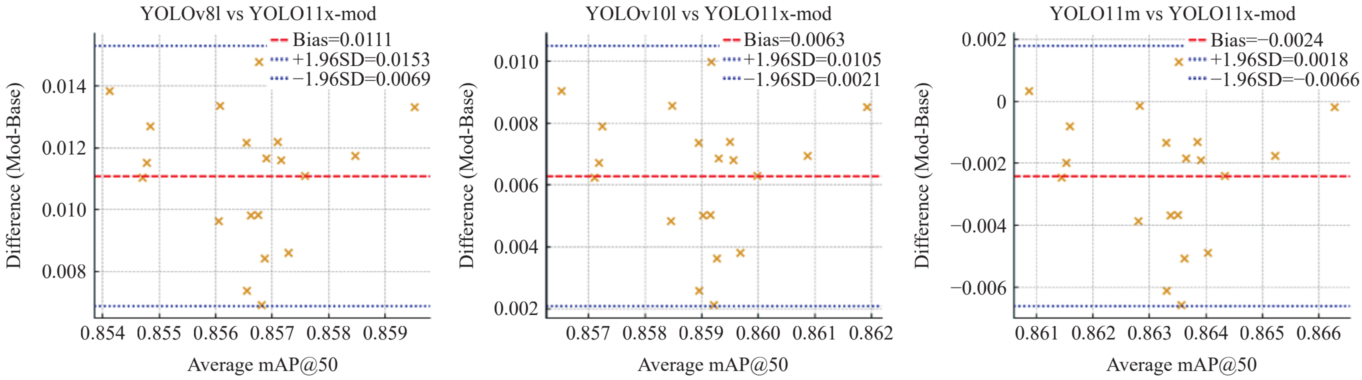


Figure 8 Diagram of the Bland-Altman analysis

4 Discussion

This study demonstrates the effectiveness of the YOLO11x-mod-based object detection model developed for kiwifruit detection. The results show that the model can operate with high accuracy even under challenging environmental conditions, providing a robust alternative particularly against dense foliage and variable lighting. The model achieved 81.76% precision, 83.97% recall, and an mAP@50 of 86.19%, offering a significant advantage in detecting fruits in complex natural scenes.

In the existing literature, methods for detecting kiwifruit and other fruits have often focused on specific conditions, been tested on a limited number of objects, or lacked sufficient accuracy. Most earlier studies relied on simpler YOLO versions or older architectures such as Faster R-CNN, which frequently resulted in high false positives and missed detections. Moreover, the detection of small objects and overlapping instances has often been reported with weak results. Our study directly addressed these shortcomings through hyperparameter strategies such as AdamW optimization, low weight decay, LR=0.005, and the dropout-mosaic combination, achieving meaningful improvements particularly in small object detection compared to previous models. Thus, the methodological originality lies not only in the chosen hyperparameter settings but also in their holistic design to overcome the fundamental limitations reported in the literature.

The computational efficiency of the proposed model was also evaluated in detail. YOLO11x-mod operates with approximately 72.5 M parameters, 272 GFLOPs, and 42 FPS. This configuration achieves higher mAP@50:95 values than similar large-scale models such as YOLOv8x and YOLOv10x, while still maintaining the real-time processing threshold. Therefore, the model stands out not only in terms of accuracy but also in the balance between computational cost and speed, distinguishing it from existing solutions in the literature. This finding highlights the reliability and practicality of our model as an alternative for agricultural automation applications requiring high accuracy.

From the perspective of real-world applications, the processing speed of 42 FPS is sufficient for real-time operation in agricultural robots. However, broader field tests are required to further assess the effects of shadowing, dense leaf coverage, and variable lighting conditions. In addition, the model's large parameter size and GFLOPs requirements may limit its direct deployment on low-power embedded devices. At this point, adapting the developed hyperparameter strategies to lighter YOLO variants or combining them with model compression techniques could facilitate future field deployment.

5 Conclusions

The evolution of sophisticated object detection models such as the YOLO11x-mod variant is notable in the automated gathering of activities such as fruit picking. By evading issues such as dense undergrowth, light variability, and crop brittleness on fruits such as kiwifruit, the models have been shown to be capable of boosting efficiency and accuracy while working in challenging conditions. Advanced hyperparameter optimization and data augmentation methods have also made these models more robust to adapt for real-world applications. While there is a dramatic boost in performance, real-world practical problems remain, especially model transfer to novel environments and model adaptation to systems with limited resources. Solutions specific to them, like lightweight substitutes and stronger contextual learning, have promise in bridging the gaps. Effective deployment of such systems will rely on continuous research, collaboration between technology developers and agricultural practitioners, and scalability awareness. Object detection technologies can shape agricultural operations, improve productivity, reduce labor dependency, and promote sustainable agriculture by filling the gap between theory and practice. The YOLO11x-mod developed in this study is not merely a model that fills the methodological gaps in the literature. It also provides a method that combines high accuracy, acceptable speed, and practical applicability, thereby making a strong contribution to agricultural automation and precision farming applications. This study sets a new benchmark in the field of agricultural object detection and offers guidance for future research both at the technical and application levels. This research paves the way for a new generation of precision agriculture, connecting technological innovation and growing worldwide demand for productive and sustainable food production.

Acknowledgements

The authors recognize the effort and contributions of members of the research team and technical staff in conceptualizing and working on this research. Special acknowledgement is extended to agricultural experts and orchard operators for granting access to the kiwifruit data as well as valuable information about the harvest problems. The authors are thankful for the facilitation provided by their respective organizations in making resource and infrastructure provision possible that could support this work. Finally, gratitude is expressed to the co-authors and reviewers whose inputs and recommendations immensely added to the quality of this work.

[References]

[1] Zhang Z Y, Zeng S C, Ji T K, Cao M K, Guo W C. Generation of fruit's spectra with hundreds of wavelengths from obtained multi-spectra and spectral application using deep learning. Computers and Electronics in Agriculture, 2023; 210: 107882.

[2] Wu K, Jia Z C, Duan Q F. The detection of kiwifruit sunscald using spectral reflectance data combined with machine learning and CNNs. Agronomy, 2023; 13(8): 2137.

[3] Wan C, Yue R, Li Z F, Fan K, Chen X K, Li F L. Prediction of kiwifruit sweetness with Vis/NIR spectroscopy based on scatter correction and feature selection techniques. Applied Sciences, 2024; 14(10): 4145.

[4] Wang L. Improve yield prediction with UAS-based leaf area index estimation and a hybrid machine learning-and process-based model. PhD dissertation. Lincoln: The University of Nebraska-Lincoln. 2022; 114p.

[5] Wang L Z, Gao R, Li C C, Wang J, Liu Y, Hu J Y, Yue J B. Mapping soybean maturity and biochemical traits using UAV-based hyperspectral images. Remote Sensing, 2023; 15(19): 4807.

[6] Liu J F, Liu Z B. YOLOv5s-BC: an improved YOLOv5s-based method for real-time apple detection. Journal of Real-Time Image Processing, 2024; 21(3): 88.

[7] Zhang J L, Su W H, Zhang H Y, Peng Y. SE-YOLOv5x: An optimized model based on transfer learning and visual attention mechanism for identifying and localizing weeds and vegetables. Agronomy, 2022; 12(9): 2061.

[8] Yang W J, Ma X X, Hu W C, Tang P J. Lightweight blueberry fruit recognition based on multi-scale and attention fusion NCBAM. Agronomy, 2022; 12(10): 2354.

[9] Su F, Zhao Y P, Shi Y X, Zhao D, Wang, G H, Yan Y F, et al. Tree trunk and obstacle detection in apple orchard based on improved YOLOv5s model. Agronomy, 2022; 12(10): 2427.

[10] Wu Y Q, Sun Y B, Zhang S Q, Liu X, Zhou K, Hou J L. A size-grading method of antler mushrooms using YOLOv5 and PSPNet. Agronomy, 2022; 12(11): 2601.

[11] Rong J C, Fu J, Zhang Z Q, Yin J L, Tan Y Z, Yuan T, et al. Development and evaluation of a watermelon-harvesting robot prototype: Vision system and end-effector. Agronomy, 2022; 12(11): 2836.

[12] Xie J X, Peng J J, Wang J X, Chen B H, Jing T W, Sun D Z, et al. Litchi detection in a complex natural environment using the YOLOv5-Litchi model. Agronomy, 2022; 12(12): 3054.

[13] Fu X, Li A, Meng Z, Yin X, Zhang C, Zhang W, et al. A dynamic detection method for phenotyping pods in a soybean population based on an improved YOLO-v5 Network. Agronomy, 2022; 12(12): 3209.

[14] Wang C, Wang H, Han Q Y, Zhang Z G, Kong D D, Zou X J. Strawberry detection and ripeness classification using YOLOv8+ model and image processing method. Agriculture, 2024; 14(5): 751.

[15] Wang Y, Rong Q, Hu C. Ripe tomato detection algorithm based on improved YOLOv9. Plants, 2024; 13(22): 3253.

[16] Zhou Z X, Song Z Z, Fu L S, Gao F F, Li R, Cui Y J. Real-time kiwifruit detection in orchard using deep learning on android™ smartphones for yield estimation. Computers and Electronics in Agriculture, 2020; 179: 105856.

[17] Lim J, Ahn H S, Nejati M, Bell J, Williams H, MacDonald B A. Deep neural network-based real-time kiwi fruit flower detection in an orchard environment. arXiv preprint, 2020; arXiv: 2006.04343. doi: 10.48550/arXiv.2006.04343.

[18] Ulutas E G, Altın C. Kiwi fruit detection with deep learning methods. International Journal of Advanced Natural Sciences and Engineering Researches, 2023; 7(7): 39–45.

[19] Pawar S E, Surana A V, Sharma P, Pujeri R. Fruit disease detection and classification using machine learning and deep learning techniques. International Journal of Intelligent Systems and Applications in Engineering, 2024; 12(4s): 440–453.

[20] Xia Y, Nguyen M, Yan W Q. A real-time kiwifruit detection based on improved YOLOv7. In: Image and Vision Computing. IVCN 2022. Lecture Notes in Computer Science, 2023; pp.48–61. doi: 10.1007/978-3-031-25825-1_4.

[21] Dewi C, Thiruvady D, Zaidi N. Fruit classification system with deep learning and neural architecture search. arXiv preprint, 2024; arXiv: 2406.01869. doi: 10.48550/arXiv.2406.01869.

[22] Selva Kumari R S, Gomathy V. Fruit shop tool: Fruit classification and recognition using deep learning. Agricultural Engineering International: CIGR Journal, 2023; 25(2): 312–321.

[23] Ma L, He Z, Zhu Y T, Jia L S, Wang Y C, Ding X T, et al. A method of grasping detection for kiwifruit harvesting robot based on deep learning. Agronomy, 2022; 12(12): 3096.

[24] Banerjee D, Kukreja V, Hariharan S, Sharma V. Fast and accurate multi-classification of kiwi fruit disease in leaves using deep learning approach. In: 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA), Uttarakhand: IEEE, 2023; pp.131–137. doi: 10.1109/ICIDCA56705.2023.10099755.

[25] Ebrahimi S, Pourdarbani R, Sabzi S, Rohban M H, Arribas J I. From harvest to market: Non-destructive bruise detection in kiwifruit using convolutional neural networks and hyperspectral imaging. Horticulturae, 2023; 9(8): 936.

[26] Elsayed S, Gala H, Abd El-baki M S, Maher M, Elbeltagi A, Salem A, et al. Hyperspectral technology and machine learning models to estimate the fruit quality parameters of mango and strawberry crops. PloS One, 2025; 20(2): e0313397.

[27] Niu Z J, Deng J T, Zhang X., Zhang J, Pan S J, Mu H T. Identifying the branch of kiwifruit based on unmanned aerial vehicle (UAV) images using deep learning method. Sensors, 2021; 21(13): 4442.

[28] Song Z Z, Zhou Z X, Wang W Q, Gao F F, Fu L S, Li R, et al. Canopy segmentation and wire reconstruction for kiwifruit robotic harvesting. Computers and Electronics in Agriculture, 2021; 181: 105933.

[29] Yang Y, Su L J, Zong A Y, Tao W H, Xu X P, Chai Y X, et al. A new kiwi fruit detection algorithm based on an improved lightweight network. Agriculture, 2024; 14(10): 1823.

[30] Zhao G R, Dong S, Wen J, Ban Y C, Zhang, X W. Selective fruit harvesting prediction and 6D pose estimation based on YOLOv7 multi-parameter recognition. Computers and Electronics in Agriculture, 2025; 229: 109815.

[31] Rodríguez-Álvarez J L, García-Alcaraz J L, Vidal R P, Cuevas-Jacques R, Díaz-Reza J R. Noninvasive system for weight estimation in cactus crops: A YOLOv5-decision tree approach based on interval type-2 fuzzy sets. Chemometrics and Intelligent Laboratory Systems, 2024; 245: 105064.

[32] Pesce L. Machine learning for predicting grape quality using spectral imaging techniques. PhD dissertation, Politecnico di Torino, 2024; 101p.