

Article

Convolutional Neural Networks for Detecting White Grape Bunches in High-Density Vineyards

Valeriano Méndez Fuentes ¹, Lourdes Lleó ², Pilar Barreiro Elorza ^{2,*}, Abraham Tamargo-Vinces ²,
Wilson Valente Da Costa Neto ^{2,3}, Adolfo Moya González ², Pablo Guillén ² and Pilar Baeza ³

¹ Department Applied Mathematics, Technical University of Madrid, 28040 Madrid, Spain; valeriano.mendez@upm.es

² LPF_TAGRALIA, Department Agroforestry Engineering, Technical University of Madrid, 28040 Madrid, Spain; lourdes.lleo@upm.es (L.L.); wilsonneto@unipampa.edu.br (W.V.D.C.N.); adolfo.moya@upm.es (A.M.G.)

³ TAPAS, Department Plant Production, Technical University of Madrid, 28040 Madrid, Spain; pilar.baeza@upm.es

* Correspondence: pilar.barreiro@upm.es

Abstract

This study addresses the challenge of detecting white grape bunches (*Vitis vinifera* L.) in high-density vineyard canopies, a critical task for precision viticulture and yield estimation. Traditional statistical and image-processing methods have struggled to cope with occlusion issues. In this work, more than 200 field RGB images were collected at La Bergonza (Toledo, Spain) and expanded through data augmentation. Several preprocessing strategies were evaluated to enhance bunch visibility. Different convolutional neural network (CNN) architectures were compared, with YOLOv8 outperforming Mask R-CNN in terms of both accuracy and efficiency. YOLOv8, trained for up to 100 epochs on equalized and augmented datasets, achieved outstanding performance, with 84.9% precision, 72.6% recall, and an mAP@0.5 of 83%, far surpassing Mask R-CNN (17% precision and 26% recall). The model successfully detected partially occluded grape bunches, including some that were not visible to human experts, and outperformed previous studies that relied on controlled backgrounds or artificial lighting. The results demonstrate that combining RGB equalization with data augmentation significantly improves detection performance. These findings highlight the potential of deep learning and low-cost RGB imaging systems to enable automated and scalable solutions for yield estimation and canopy analysis. In conclusion, YOLOv8 emerges as a promising tool for accurate grape bunch detection under real field conditions, effectively overcoming previous technological limitations.

Keywords: deep learning; computer vision; precision agriculture; grape yield analysis



Academic Editor: Aichen Wang

Received: 24 March 2026

Revised: 29 April 2026

Accepted: 7 May 2026

Published: 13 May 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The evaluation of grape yield and quality is a central challenge in precision viticulture, as expert-based assessments are labor-intensive and often unreliable when only a limited number of reference plots are used. Among quality-related traits, grape bunch compactness is particularly important due to its strong influence on yield, grape and wine quality, and susceptibility to fungal diseases. Foundational work by Tello and Ibáñez [1] provided a comprehensive multi-season analysis of morpho-agronomic variables affecting compactness, while a subsequent review [2] identified expert-validated compactness indices—such

as CI-12, CI-18, and CI-19—highlighting the need for objective, image-based alternatives capable of assessing compactness at the ripeness stage.

Early computer vision approaches addressed this need through pixel-based classification of vineyard RGB images. Herrero-Langreo et al. [3] demonstrated that vineyard elements such as bunches, leaves, and canopy porosity could be accurately segmented using the Mahalanobis distance. Correa et al. [4] further compared several fuzzy clustering techniques across multiple color spaces, showing that image resolution reduction can significantly improve computational efficiency without degrading accuracy, although some clustering methods produced overlapping bunches and classification errors. These limitations were later mitigated by Correa et al. [5], who used Gustafson–Kessel fuzzy c-means to generate optimal centroids for K-means clustering, significantly improving segmentation accuracy.

Building on segmentation-based methods, supervised classification approaches were developed to estimate canopy traits and yield. Diago et al. [6] achieved high classification accuracy for leaves and bunches using Mahalanobis-based classifiers and reported strong correlations between image-derived variables and ground-truth leaf area and yield. Importantly, they observed that vineyard defoliation positively influenced estimation accuracy. This observation was reinforced by Íñiguez et al. [7], who demonstrated that increasing bunch occlusion severely degrades yield prediction performance, with model accuracy improving dramatically under partial or total defoliation.

More recently, deep learning approaches have been introduced to address these limitations. Íñiguez et al. [8] employed YOLO-based models with varying defoliation levels and showed that detection accuracy improves as occlusion decreases, although performance deteriorates in dense canopies. These challenges motivated technology transfer efforts, including patents describing automated estimation of vineyard porosity [9,10] and grape bunch compactness in winery environments [11,12], demonstrating strong agreement with traditional evaluation methods and highlighting the practical relevance of image-based solutions.

At a broader level, the application of artificial intelligence to grape yield estimation has expanded rapidly. Object detection frameworks such as YOLO, Faster R-CNN, SSD, and RetinaNet have been widely explored, although difficulties in bounding box localization under complex canopy conditions persist [13]. Comprehensive reviews, such as that of Mohimont et al. [14], emphasize the growing role of CNN-based object detection and semantic segmentation in vineyard monitoring using standard RGB imagery, without reliance on specialized sensors.

Recent studies have further evaluated advanced deep learning models under field conditions. Casado-García et al. [15] demonstrated the effectiveness of semantic segmentation architectures (e.g., DeepLabV3+, MANet) for grape bunch identification using RGB-D imagery, while Palacios et al. [16] incorporated nighttime illumination and SegNet-based models for berry-level yield estimation. Despite methodological advances, a comprehensive review by Huang et al. [17] highlights that data scarcity, multi-scale detection, and, above all, severe occlusion remain the most critical challenges for deep learning-based crop object detection.

Also, Convolutional neural networks (CNNs) have increasingly become central to computer vision in broader applications in precision viticulture, as reflected in several recent contributions. One of the earlier and influential studies by Aguiar et al. (2021) [18] proposed a CNN-based approach for vine trunk detection, using Single Shot Multibox Detector architectures to support robotic navigation and localization in complex vineyard environments characterized by steep slopes and limited satellite coverage. Expanding on this line of research, Pacioni et al. (2025) [19] demonstrated that modern CNN-based

segmentation frameworks, particularly YOLOv8 and Mask R CNN, can reliably identify trunks and vine shoots in real time, enabling intelligent robotic pruning systems suitable for deployment on embedded GPU platforms. Together, these studies highlight the growing maturity of CNN-driven perception systems tailored to structural analysis and automation in vineyards.

Beyond vine structure and robotics, CNNs have also been widely applied to vineyard-related crop management tasks, notably weed and disease detection. A systematic review by García Navarrete et al. (2024) [20] analyzed recent CNN architectures for weed detection across multiple crops and emphasized their relevance for vineyards, where site-specific weed control is critical for sustainable management. From a broader perspective, Albahar (2023) [21] surveyed the impact of deep learning in agriculture and identified CNN-based crop monitoring, disease detection, and yield estimation as high-potential applications for specialty crops such as grapevine.

Overall, the literature shows consistent progress toward automated grape yield and quality assessment, while repeatedly identifying canopy density and occlusion as unresolved barriers. These findings underscore the need for robust deep learning approaches capable of accurately detecting grape bunches in dense vineyards under natural field conditions.

The objective of this study is to address the problem of identifying white grapes in very high-density canopies (undeafed), using RGB images. This goal goes beyond the current state of the art by making white grapes the target of segmentation.

2. Materials and Methods

2.1. Description of the Vineyard

The vineyard plot La Bergonza is located at 40.1534278° N latitude and −4.2207311° longitude (Toledo, Spain) and belongs to the González Byass winery. Established in 2018 with *Vitis vinifera* L. variety Airén (row × vine spacing of 3.3 m × 2.0 m), the vineyard is dedicated to high-intensity wine grape production, with yields ranging between 30,000 and 40,000 kg ha^{−1}. The single-curtain training system results in a very dense grape bunch zone, characterized by compact bunches and a porous canopy structure. Taking into account a vine spacing of 2 m along the row, the productivity of the vine is notably high, reaching between 19.8 and 26.5 kg per vine, with 77 to 103 grape bunches per vine. The bunches have a median weight of 0.256 kg, and the grapes exhibit an average soluble solids content of 15.8 ± 4.0 °Brix. Harvest was carried out on 29 September 2023.

2.2. Image Acquisition

The imaging system used in this study was a Sony Alpha 77 (α77, Sony Corporation, Japan) digital interchangeable lens camera, based on an APS C format sensor and Sony's Single Lens Translucent (SLT) architecture. The camera was selected for its high spatial resolution, fast acquisition rate, and continuous phase detection autofocus, which are advantageous for scientific and technical imaging applications.

The α77 incorporates a 24.3 megapixel APS C Exmor CMOS sensor with approximate dimensions of 23.5 × 15.6 mm. This sensor enables high-resolution image acquisition with fine spatial detail and supports a native sensitivity range of ISO 100–16000, facilitating imaging across a wide range of illumination conditions. Signal processing is performed using Sony's BIONZ image processor, which provides real-time image computation, noise reduction, color processing, and high-throughput data handling during continuous capture.

The Sony α77 has a fixed translucent mirror, which partially reflects incident light toward a dedicated phase detection autofocus module while transmitting the remaining light to the image sensor. This SLT configuration enables continuous phase detection autofocus

during still image capture, high-speed continuous shooting, and video recording. Unlike conventional digital single lens reflex (DSLR) systems, this design eliminates mechanical mirror motion, thereby reducing shutter lag and mechanical vibration while maintaining autofocus functionality at all times.

The camera supports continuous image acquisition at rates up to 12 frames per second at full sensor resolution. This capability allows reliable capture of fast or transient phenomena and is suitable for time-resolved imaging and motion analysis. Autofocus performance is provided by a 19-point phase detection autofocus system, including multiple cross-type sensors, which improves robustness and accuracy in subject tracking compared with single-axis detection systems.

Sony α 77 provides image composition and real-time monitoring via a 2.36-million-dot electronic viewfinder (EVF), offering 100% image coverage. The EVF enables immediate visualization of exposure, focus, white balance, and field-depth parameters before image acquisition.

In addition to still image acquisition, the Sony α 77 supports Full High Definition (1920×1080) video recording with continuous phase detection autofocus. Manual control of exposure parameters during video capture allows reproducible acquisition settings, enabling the camera to be used in hybrid workflows combining still imaging and video documentation.

The images were taken from the center of the inter-vine lines centered about the vine-wire central conduction. They were recorded between 8:00 and 13:00 to increase the variability of light as the sun rose from beneath the vine towards the highest position on the horizon.

2.3. Image Pre-Processing

A total of 236 images were captured, of which 177 were used for the training set and 59 for the validation set. Different equalization techniques were applied to train multiple detection models to evaluate the impact of each approach on training quality and model accuracy MATLAB R2024a (The MathWorks, Inc., Natick, MA, USA). Several histogram equalization methods were used to improve visual quality and grape bunch detection. The applied equalization techniques are illustrated in Figure 1 and include the following:

- Global RGB channel histogram equalization: Global equalization was applied independently to the three-color channels (red, green, and blue). This approach improves contrast across the color spectrum, making grape bunches more distinguishable in complex environments where background colors may be similar.
- Global greyscale histogram equalization: This technique was applied to the greyscale version of the images to improve overall contrast, particularly in images captured under unfavorable lighting conditions. Global equalization helps distribute intensity levels more evenly, facilitating better discrimination between grape bunches and background elements.
- CLAHE (contrast-limited adaptive histogram equalization) applied to the green channel: CLAHE was applied specifically to the green channel, which contains a significant proportion of relevant information in vegetation images. This method enhances local contrast while limiting excessive contrast amplification in homogeneous regions.

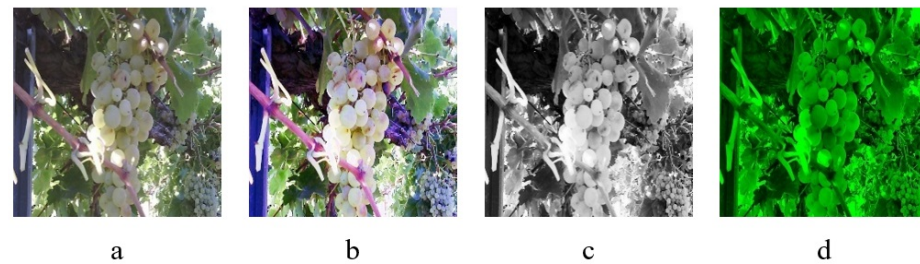


Figure 1. Examples of the equalization processes: (a) original image, (b) RGB histogram equalization, (c) greyscale image with CLAHE, and (d) RGB channel equalization.

2.4. Methodology Using CNN

Computer vision is a subfield of machine learning that enables the automatic interpretation and identification of images. Although simple image classification tasks can be addressed using multilayer perceptrons [22], convolutional neural networks (CNNs) [23] enable a more general and robust image classification. The major milestones in CNNs development include the AlexNet architecture [24], followed by VGGNet [25], and later ResNet, which introduced residual learning [26].

The Region-based Convolutional Neural Network (R-CNN) method [27] initially fine-tunes a convolutional network using a logarithmic loss function, and subsequently refines the extracted features using support vector machines (SVMs). These SVMs act as object detectors and enable the regression of bounding boxes. Fast R-CNN [28] improves both performance and accuracy by introducing a single-stage training approach based on a multi-task loss function.

Fast R-CNN operates on the full image along with a set of object proposals. The image is processed through multiple convolution and max-pooling layers to generate a feature map. For each object proposal, a region-of-interest (RoI) pooling layer extracts a fixed-length feature vector from the feature map. This vector is then passed through fully connected layers to perform both bounding box regression and object classification. A subsequent evolution, Mask R-CNN [29], extends this architecture by adding a branch that predicts a pixel-level segmentation mask for each detected object, in addition to the regression and classification of the bounding box.

YOLO (You Only Look Once) [30] is a single-stage object detection network that directly predicts bounding boxes and class probabilities for objects in a 2D scene. Several versions of this architecture have been developed, culminating in YOLOv8, which is used in this work [31]. YOLOv8 predicts bounding boxes and class probabilities without requiring a separate region proposal network. It adopts a center-based, anchor-free detection approach and implements pseudo-ensemble and pseudo-supervision strategies, which involve training multiple models to generate more diverse predictions, thereby improving detection accuracy and robustness.

The original images have a resolution of 1920×1080 RGB, while the input to YOLOv8 is 640×640 RGB; the annotation was made with proprietary software. The augmentation was made manually, increasing the dataset to 619 images.

YOLOv8 uses the CSPDarknet53 backbone network (Figure 2), which is conceptually similar to GoogLeNet [32], for the extraction of features, along with a neck and head architecture. The main architectural updates compared to earlier YOLO versions include replacement of the CSP layer with the C2f module; substitution of 6×6 convolutions with 3×3 convolutions; replacement of 1×1 convolutions with 3×3 convolutions in bottleneck layers; removal of the 10th and 14th convolutional layers; introduction of a decoupled detection head; and elimination of the branch with identified objects.

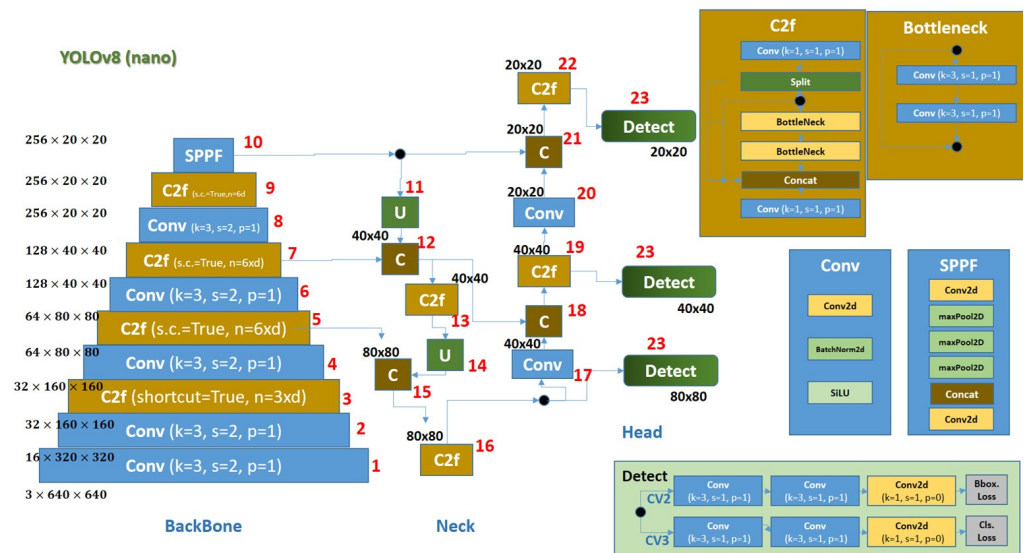


Figure 2. Overview of the YOLOv8 architecture. On the left, the convolutional backbone network is composed of Conv and C2f layers. The backbone starts with three RGB channels of spatial dimension 640 and outputs 512 channels of dimension 20. At two intermediate stages and at the final stage of the backbone, feature maps are forwarded to the neck and detection head. The detection component enables the extraction of bounding boxes, class predictions, and segmentation masks. The symbol U denotes an upsampling operation used to increase the resolution of the feature map, while C represents a concatenation operation that merges two feature maps. The numbers in red indicate the execution order of each layer. The dots indicate the concatenation of feature maps of equal dimension. The arrows indicate data input and output between layers. For example, the triplets $3 \times 640 \times 640$ refer to the dimensions in channels, height, and width of the feature maps. Another example, whenever pairs of 80×80 appear, the height and width dimensions of the feature maps are highlighted.

The convolutional blocks consist of a Conv2D layer, Batch Normalization, and SiLU activation. The bottleneck layers resemble residual blocks and comprise two convolutional blocks linked by a shortcut connection. The C2f module begins with a convolutional block that branches either into two bottleneck layers or directly into a concatenation layer. The Spatial Pyramid Pooling Fast (SPPF) block includes a convolutional layer, three max-pooling layers, and a final concatenation step followed by another convolutional layer.

The backbone network consists of ten blocks (Figure 2), alternating Conv and C2f layers, and terminating with an SPPF block. An RGB image of size 640×640 is input to the backbone, producing 512 feature maps of size 20×20 in the output. The backbone provides three outputs to the neck: one from the final SPPF layer with a spatial resolution of 20×20 , and two intermediate outputs from the fifth and seventh blocks with resolutions of 80×80 and 40×40 , respectively.

The neck includes vertical connections with upsampling operations that increase the feature map resolution from 20 to 40 and 80. These maps are laterally connected to the head, which also incorporates vertical connections using convolutional blocks to progressively reduce the spatial resolution from 80 to 40 and then to 20. This process leads to three Detect blocks operating at resolutions of 20, 40, and 80. Each detection block branches into two paths composed of a pair of convolutional layers with a 3×3 kernel, followed by a Conv2D layer with a 1×1 kernel. One branch is responsible for bounding box regression, while the other estimates class probabilities. In addition, YOLOv8 includes a segmentation variant, YOLOv8-Seg, which incorporates two semantic segmentation heads.

The training workflow (Figure 3) begins with the loading of the image files and their corresponding annotations. The images are optionally equalized, resulting in tensors of

dimensions (3, w, h), where w and h represent the width and height of the image across the three RGB channels. Annotation files provide target information, including object classes, bounding boxes, and, when applicable, segmentation masks for Mask R-CNN or YOLOv8 segmentation models.

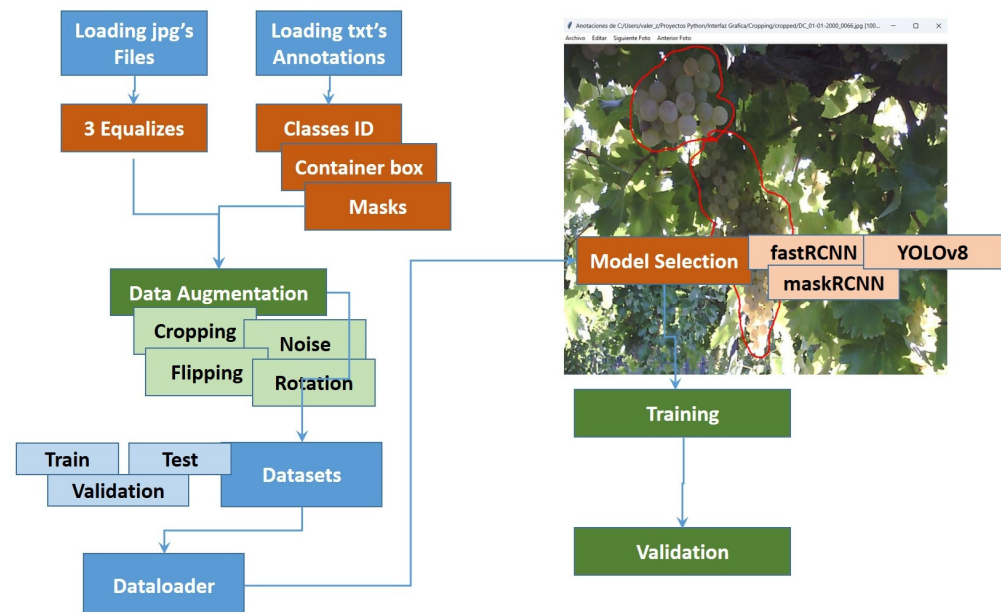


Figure 3. Training process. The upper-right section illustrates how the information in the data set is stored, including image files and the corresponding annotation text files. Annotations may consist of either bounding box coordinates or polygon vertices that define object masks. The polygon lines that define the two-cluster mask are shown in red on the image. The data augmentation process—comprising cropping, noise addition, flipping, and rotation—increases the number of samples in the data set to its final size. The augmented data set is then divided into training and testing subsets for model training, as well as a validation subset for model evaluation, all managed through a data loader. The training process can be performed using any of the implemented models, including Fast R-CNN, Mask R-CNN, or YOLOv8.

The subsequent step involves data augmentation, which generates an expanded dataset that is then shuffled and divided into training, validation, and testing subsets. A data loader is used to manage access to the data set during training and validation. Data augmentation techniques include random noise addition, cropping, flipping, and rotation, all of which increase the diversity and size of the training data.

In this work, the YOLOv8 framework was trained for up to 100 epochs. Several evaluation metrics were considered, including the model confidence score threshold; true positives (TP), false positives (FP), and false negatives (FN); total predicted boxes (PB); total ground-truth boxes (RB), including those in the training dataset; precision (TP/(TP + FP)); recall (TP/(TP + FN)); and the maximum confidence score among predicted bounding boxes.

The database of 236 images was acquired in the field for this study. It was augmented manually using Pytorch 2.5.1-cuda-11.8 up to 619 images, which were split randomly into 495 for training and 124 for validation. The GPU processor is NVIDIA Corporation TU117GL [T1000 8Gbytes]. The augmentation procedures are random horizontal and vertical flips, and random rotation with 89 and 91 degrees. The hardware employed is 12th Gen Intel® core™ i7-12700.

The number of training epochs influences the degree of model fitting. During each epoch, the network parameters are updated by comparing the predicted output with the ground truth. In the experiments conducted in this paper, Mask R-CNN was trained for

100 epochs, while YOLOv8 was initially trained for 50 epochs, and subsequently extended to a total of 100 epochs (Figure 4).



Figure 4. On the (left), an image illustrates the annotations used for Mask R-CNN, which consist of closed polygonal lines in red that define the object masks required by the model. On the (right), an image shows the annotations used for YOLOv8, which are based on rectangular bounding boxes in yellow.

2.5. Metrics

- IoU (Intersection over Union): measures the overlap between the predicted bounding box or mask and the ground truth and is defined in Equation (1).

$$\text{IoU} = \frac{\text{Intersection Area}}{\text{Union Area}} \quad (1)$$

If $\text{IoU} = 1$, then the prediction is perfect, while an $\text{IoU} = 0$ indicates no overlap; if for a bounding box IoU is greater than the threshold, it is considered a True Positive (TP), and if it is below, it is considered a False Positive (FP); finally a False Negative (FN) is a bounding box whose IoU is lower than the threshold and that it exist in the ground-truth and a False Positive (FP) is a bounding box whose IoU is higher than the threshold but it does not exist in the ground-truth. In this paper, a range between 0.35 and 0.65 for the threshold is evaluated.

Based on the above evaluated range of IoU, the following metrics can be defined, aggregated across the set of bounding box detections (Equations (2)–(6)).

- Accuracy (ACC): the ground-truth values over all data.

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \quad (2)$$

- Precision: of all predictions, indicates the correct fraction, using an IoU threshold of 0.5

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3)$$

- Recall: of all real objects, indicates the fraction correctly detected. In the following formula, false negatives (FN) are real objects that were not predicted or whose prediction fell below the threshold of 0.5

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4)$$

- F1-score: is the harmonic mean between precision and recall

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

- AP (Average Precision): obtained by integrating the precision-recall (P-R) curve for a class. It is a value between 0 and 1 or between 0% and 100%. The P-R curve is obtained by changing the threshold of IoU values. When the IoU threshold is very high, the number of FN increases dramatically, and FP is reduced, which increases recall and reduces precision. Conversely, if the threshold of IoU is low, FN is reduced and FP increases, lowering recall and increasing precision (Figure 5). The point highlighted in blue in Figure 5 is an example of the computation of the confusion matrix.

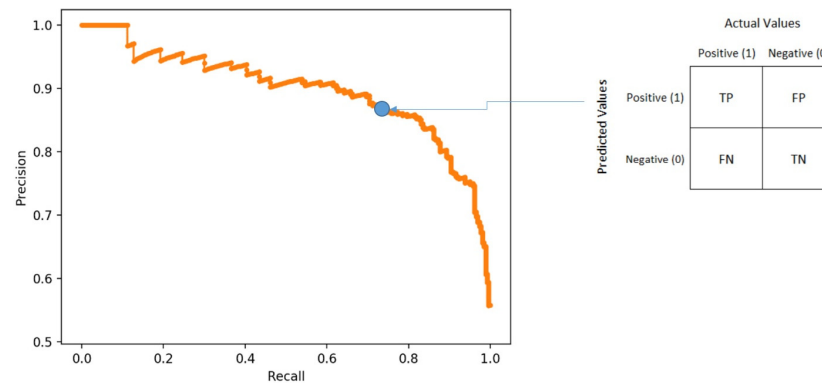


Figure 5. By changing the IoU threshold value, all the containing boxes are evaluated, and based on the resulting value (TP, FP, FN, and TN), the confusion matrix is obtained with the count of all of them (**right**). Using the data from the confusion matrix, the precision and recall values are obtained by drawing the PR-Recall graph, on the (**left**) in orange. The integral under the PR-Recall graph allows one to obtain the AP value. Note that the area of the curve is contained in a square with side size 1, so it will have values $0 < AP < 1$. The point highlighted in blue is an example of the computation of the confusion matrix.

- mAP (mean Average Precision): average of the AP across all classes

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (6)$$

- mAP@0.5: average of the APs with an IoU threshold of 0.5. If the $IoU \geq 0.5$, the detection is considered correct (TP). If the $IoU < 0.5$, it is considered incorrect (FP).
- mAP@0.5:0.95: The AP is calculated several times, with different IoU thresholds: from 0.5 to 0.95, in steps of 0.05: 0.50, 0.55, 0.60, ..., 0.95. Finally, the AP obtained at each of these thresholds is averaged. This is a stricter and more complete metric, whereas mAP@0.5 is more permissive as it only requires a 50% overlap.
- Fitness: is a weighted average of the metrics: precision, recall, mAP@0.5 and mAP@0.5:0.95. By default, in YOLOv8, the weighting [0, 0, 0.1, 0.9] is taken respectively for the four.

3. Results

Tables 1 and 2 summarize the results of the identification of grape stalks using the Mask R-CNN approach. The experiments were conducted using 100 training epochs, with 177 images in the training set and 59 images in the testing set. Several models were trained using different Intersection over Union (IoU) and model score thresholds, resulting in varying values of true positives (TP), false positives (FP), false negatives (FN), total predicted boxes (PB), total real boxes (RB), total real boxes in the training dataset, precision ($TP/(TP + FP)$), recall ($TP/(TP + FN)$) and maximum confidence score of predicted boxes.

Table 1. Results obtained with the Mask R-CNN model trained for 100 epochs using 177 training images and 59 testing images, for different IoU thresholds, including true positives (TP), false positives (FP), false negatives (FN), precision and recall.

Score Threshold	True Positives (TP)	False Positives (FP)	False Negatives (FN)	Precision TP/(TP + FP)	Recall TP/(TP + FN)
0.35	67	653	101	0.093	0.399
0.40	57	540	113	0.095	0.335
0.45	50	515	116	0.088	0.301
0.50	42	424	119	0.090	0.261
0.55	65	430	129	0.130	0.336
0.60	38	285	120	0.117	0.241
0.65	40	197	111	0.170	0.265

Table 2. Results obtained with the Mask R-CNN model trained for 100 epochs, showing the influence of model score thresholds on the total number of predicted boxes, the total real boxes in the validation dataset, and the total real boxes in the training dataset.

Score Threshold	Max Score in Predict Box	Total Predicted Boxes (PB)	Ground-Truth Boxes (Validation)	PB/GTB Validation (%)	Ground-Truth Boxes (Train)
0.35	0.998	720	161	447%	506
0.40	0.995	597	167	357%	500
0.45	0.996	565	161	351%	506
0.50	0.998	466	159	293%	508
0.55	0.991	495	187	265%	480
0.60	0.991	323	155	208%	512
0.65	0.998	237	147	161%	520

The best performance was obtained using a score threshold of 0.65, which yielded a maximum predicted box confidence of 0.998 and minimized false positives and false negatives (197 and 111, respectively). However, the results also revealed a general tendency towards over-detection of grape bunches, with an identification rate of 161%, corresponding to 237 labeled regions compared to 147 actual grape bunches (Table 3, score threshold = 0.65). Conversely, very poor performance was observed for a score threshold of 0.35, with only 67 true positives, 653 false positives, and 101 false negatives. At this threshold, precision and recall dropped to 0.093 and 0.399, respectively, demonstrating the sensitivity of the Mask R-CNN model to score threshold selection.

Table 3. Results for the YOLOv8 model trained for 50 epochs (first row) and 100 epochs (second row), using 495 training images and 124 validation images.

Model	Precision	Recall	mAP@0.5	mAP@0.5:0.95	Fitness
CLAHE	0.746	0.654	0.719	0.331	0.370
	0.799	0.694	0.776	0.405	0.442
RGB eq.	0.819	0.684	0.765	0.374	0.413
	0.850	0.747	0.830	0.451	0.489
GRAY eq.	0.753	0.612	0.696	0.327	0.364
	0.822	0.677	0.766	0.397	0.434
RGB Aug.	0.774	0.717	0.776	0.379	0.419
	0.849	0.726	0.839	0.465	0.503

Figure 6 presents an example of grape bunch identification using different Mask R-CNN models. Some configurations successfully detect grape bunches that are difficult to

distinguish with the naked eye, highlighting the potential of deep learning techniques to identify partially occluded fruit.

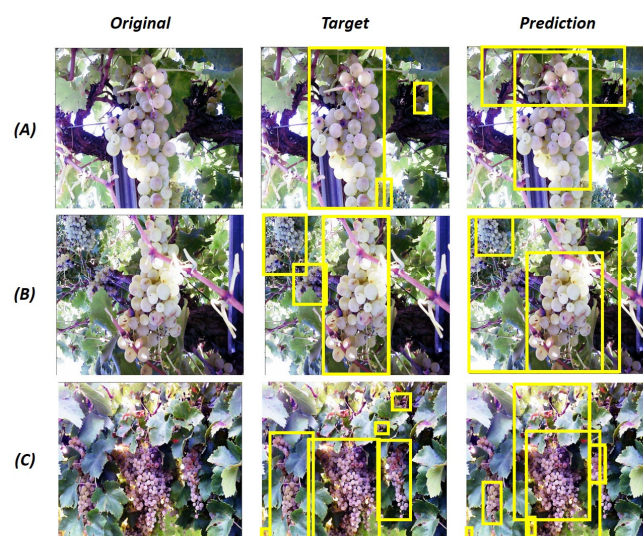


Figure 6. Examples ((A)-top, (B)-middle, (C)-low) of grape bunches identified using multiple Mask R-CNN models, including cases where grape bunches are barely visible to human observers; columns left, middle and right refer respectively to the original image, the target and the prediction model. The delimiting boxes, which identify each bunch, are represented with yellow rectangles for the targets and predictions.

Table 3 reports the performance of the YOLOv8 model for grape bunch detection using different image equalization techniques and training durations of 50 and 100 epochs. Among the approaches evaluated, the equalization of the RGB histogram and the combined filtering method achieved the highest performance in terms of mean Average Precision (mAP). RGB equalization provided the highest overall accuracy, while the combined filter offered a better balance between precision and recall, resulting in the best overall fitness score.

Table 3 includes several performance metrics, namely fitness, recall, mAP@0.5, and mAP@0.5–0.95. The results indicate that the best-performing model corresponds to the augmented dataset combined with RGB equalization. All evaluated models achieved precision values above 75%, while recall values exceeded 61%, confirming the robustness of the YOLOv8 architecture for grape bunch detection under challenging conditions.

In object detection tasks, accuracy is generally not considered a reliable evaluation metric due to the typically large and undefined number of true negatives (TN). However, by excluding the TN component, a modified accuracy metric can be derived from the confusion matrix, as presented in Table 4. Using this approach, the highest accuracy was achieved by the model trained on the augmented data set with RGB equalization, further supporting its superior performance.

Table 4. Accuracy values were calculated from the confusion matrix data for the different models evaluated.

Model	TP	FP	FN	ACC
CLAHE	651	201	222	61%
RGB eq.	694	197	165	66%
GRAY eq.	650	198	223	61%
RGB Aug.	720	208	153	67%
Mask R-CNN	320	8910	615	3%

4. Discussion

This paper highlights the effectiveness and robustness of the YOLOv8 deep learning model for detecting white grape bunches in real vineyard conditions—characterized by natural daylight, dense canopies, and significant occlusion—and compares its performance with previous state-of-the-art methods such as Mask R-CNN.

This study demonstrates that the results obtained using YOLOv8 outperform those achieved with Mask R-CNN. This performance advantage can be attributed to the more recent architecture of YOLOv8, which incorporates advanced loss functions, improved integration of batch normalization and regularization strategies, and lighter but more efficient backbone networks.

This study demonstrates that the results obtained using YOLOv8 outperform those achieved with Mask R-CNN, a performance advantage that can be attributed to the more recent architecture of YOLOv8, which incorporates advanced loss functions, improved batch normalization and regularization strategies, and lighter yet more efficient backbone networks. These architectural improvements contribute to enhanced detection accuracy, particularly in complex outdoor environments.

Several previous studies have addressed grape detection under more controlled or specialized conditions. For instance, Palacios et al. [16] employed SegNet to detect individual berries within grape bunches using active illumination under nocturnal imaging conditions. In contrast, the present study applies YOLOv8 to directly detect entire grape bunches under natural daylight and passive illumination, thereby demonstrating the robustness of the proposed approach in real vineyard environments.

The effect of occlusion has been extensively investigated in prior work. Íñiguez et al. [7] analyzed the impact of varying levels of leaf occlusion on grape bunch detection and yield estimation, evaluating linear regressions between computer vision-derived pixel counts and actual yield. Their findings showed that increased occlusion significantly degrades model performance, reinforcing the importance of addressing occlusion in vineyard image analysis, which constitutes a central focus of the present work. In a subsequent study, Íñiguez et al. [8] applied YOLOv4 to detect red grape bunches under daylight conditions using a white background to facilitate image segmentation. By contrast, the present study targets white grape bunch detection without any background manipulation. Despite operating under more challenging conditions, including strong occlusion and natural illumination, the proposed approach achieved higher precision (0.84 compared with 0.68) and comparable recall (0.73 versus 0.74), highlighting the effectiveness of YOLOv8 in highly occluded scenarios.

Other approaches have achieved higher accuracy by relying on controlled imaging conditions or alternative learning strategies. Casado García et al. [15] evaluated several deep learning models using 85 labeled images, reporting accuracies of 84.78% for DeepLabV3+ with ResNeXt50 and 85.69% for MANet with EfficientNetB3. Furthermore, the use of semi-supervised learning improved accuracy by approximately 5.6% to 6%, demonstrating its potential for agricultural imaging tasks. While the accuracy achieved in the present study (approximately 61%) is lower, it should be noted that our experiments were conducted under substantially more demanding conditions, including dense canopies, white grape varieties, natural in-field illumination, and the absence of background control. A fact that is crucial for open field robotics.

More complex architectural adaptations have also been proposed. Su et al. [13] introduced a YOLOv4-based grape bunch detection method incorporating a customized backbone network, a Bidirectional Path Aggregation Network (BiPAN) for multiscale feature fusion, and a relocated non-maximum suppression (RNMS) algorithm to improve bounding box localization. Evaluated under diverse field conditions, including varying

illumination, grape coloration, ripeness stages, and partial leaf occlusion, their method achieved a mAP of 87.7%, precision of 88.6%, recall of 78.3%, and an F1 score of 83.1%. These results slightly outperform those obtained in the present study, likely due to the more extensive architectural modifications and longer training regime.

Finally, the review by Huang et al. [17] reports results from Shen et al. [33], who achieved an average grape counting accuracy of 84.9% and a correlation coefficient of 0.9905 with manual counting. These results rely on feature extraction approaches similar to those described by Casado García et al. [15], further emphasizing the potential of deep-learning-based methods for grape detection and yield estimation when imaging conditions are well controlled. This would lead to the need for high-power active illumination under open field robotics.

5. Conclusions

1. The proposed YOLOv8-based approach demonstrates strong robustness for grape bunch detection in real vineyard conditions.
The model successfully detects white grape bunches using natural daylight and passive illumination, without any background manipulation, confirming its suitability for in-field deployment where environmental conditions are inherently variable.
2. Leaf occlusion remains a major challenge in grape bunch detection, but its impact is effectively mitigated by the proposed method.
Although increased occlusion negatively influences detection performance, the achieved precision and recall indicate that YOLOv8 can handle complex canopy structures more effectively than earlier approaches, particularly in highly occluded scenarios.
3. High detection precision is achieved despite challenging visual conditions.
The method reaches strong precision while maintaining comparable recall, demonstrating its capability to accurately localize grape bunches, even when dealing with white grape varieties, dense foliage, and the absence of controlled backgrounds.
4. The lower overall accuracy reflects the increased complexity of real-world vineyard imaging rather than model inadequacy.
The reduced performance metrics, compared with studies carried out under controlled conditions, highlight the trade-off between accuracy and real-field applicability and emphasize the importance of evaluating models under realistic operational constraints.
5. Advanced architectural modifications and longer training regimes can improve performance but may reduce deployment practicality.
While more complex network designs and extended training can yield higher accuracy, the competitive results obtained with a standard YOLOv8 configuration demonstrate a favorable balance between detection performance, computational efficiency, and ease of implementation.
6. Deep-learning-based methods show strong potential for precision viticulture applications.
The findings confirm that, even under uncontrolled and highly variable field conditions, deep learning approaches can provide reliable grape bunch detection, supporting their use in yield estimation and decision-support systems in commercial vineyard management.

Author Contributions: Conceptualization, P.B. and P.B.E.; methodology, P.B.E., A.M.G. and W.V.D.C.N., V.M.F. and A.T.-V.; validation, P.B.E., A.M.G. and P.G., formal analysis, P.B.E., V.M.F., L.L.; investigation, P.B.E., L.L., V.M.F., P.B.; resources, P.B.E., L.L., V.M.F., P.B.; data curation, V.M.F., A.T.-V.; writing—original draft preparation, V.M.F., L.L., P.B.E., and P.B.; writing—review and editing, V.M.F., L.L., P.B.E., P.B.; visualization, V.M.F., L.L. and P.B.E.; supervision, P.B.E. and P.B.; project

administration, P.B.E.; funding acquisition, P.B.E. and P.B. All authors have read and agreed to the published version of the manuscript.

Funding: Pilar Barreiro Elorza would like to acknowledge the funding of this paper by the research project RP2220280168. Adolfo Moya Gonzalez by REM2420280AMG; and Pilar Baeza by PA190000CEI3101. All private funding.

Data Availability Statement: The original data presented in the study are openly available at [<https://github.com/upmValeriano/racimosUva.git>]. Figure 4 shows the annotations defined for the maskR-CNN and YOLOv8 processes.

Acknowledgments: The authors would like to thank Miguel Tejerina, vineyard manager in from González-Byass Winery.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CLAHE	Contrast Limited Adaptive Histogram
CNN	Convolutional Neural Network
CSP	Cross Stage Partial
CSPD	CSPDarknet-53 is a convolutional neural network (CNN) backbone that integrates Cross Stage Partial (CSP) connections into the traditional Darknet-53 architecture
R-CNN	Region-Based Convolutional Region
RoI	Region of Interest
SegNet	A Deep Convolutional Encoder–Decoder Architecture for Image Segmentation
SPPF	Spatial Pyramid Pooling Fast
SSD	Single Shot Multi-Box Detector
SVM	Support Vector Machine
VGGNet	A convolutional neural network developed by the Computer Vision Group of Oxford University and Google DeepMind Laboratory
YOLO	You Only Look Once

References

1. Tello, J.; Ibáñez, J. Evaluation of indexes for the quantitative and objective estimation of grapevine bunch compactness. *Vitis-Geilweilerhof* **2014**, *53*, 9–16.
2. Tello, J.; Ibáñez, J. What do we know about grapevine bunch compactness? A state-of-the-art review. *Aust. J. Grape Wine Res.* **2018**, *24*, 6–23. [[CrossRef](#)]
3. Herrero-Langreo, A.; Barreiro, P.; Diago, M.P.; Baluja, J.; Ochagavia, H.; Tardaguila, J. Pixel classification through Mahalanobis distance for identification of grapevine canopy elements on RGB images. In Proceedings of the International Association for Spectral Imaging (IASIM-10), Dublin, Ireland, 18–19 November 2010.
4. Correa, C.; Valero, C.; Barreiro, P.; Diago, M.P.; Tardaguila, J. A comparison of Fuzzy Clustering Algorithms Applied to Feature Extraction on Vineyard. *Inteligencia Artificial: Revista Iberoamericana de Inteligencia Artificial*. 2011. Available online: https://www.academia.edu/17065134/A_Comparison_of_Fuzzy_Clustering_Algorithms_Applied_to_Feature_Extraction_on_Vineyard (accessed on 23 March 2026).
5. Correa, C.; Valero, C.; Barreiro, P.; Diago, M.P.; Tardaguila, J. Feature extraction on vineyard by Gustafson Kessel FCM and K-means. In Proceedings of the Mediterranean Electrotechnical Conference (MELECON), Yasmine Hammamet, Tunisia, 25–28 March 2012.
6. Diago, M.-P.; Correa, C.; Millán, B.; Barreiro, P.; Valero, C.; Tardaguila, J. Grapevine yield and leaf area estimation using supervised classification methodology on RGB images taken under field conditions. *Sensors* **2012**, *12*, 16988–17006. [[CrossRef](#)] [[PubMed](#)]
7. Íñiguez, R.; Palacios, F.; Barrio, I.; Hernández, I.; Gutiérrez, S.; Tardaguila, J. Impact of leaf occlusions on yield assessment by computer vision in commercial vineyards. *Agronomy* **2021**, *11*, 1003. [[CrossRef](#)]

8. Íñiguez, R.; Gutiérrez, S.; Poblete-Echeverría, C.; Hernández, I.; Barrio, I.; Tardáguila, J. Deep learning modelling for non-invasive grape bunch detection under diverse occlusion conditions. *Comput. Electron. Agric.* **2024**, *226*, 109421. [CrossRef]
9. Tardáguila Laso, M.J.; Millán Prior, B.; Diago Santamaría, B.P. Patente de Invención B1: Procedimiento para la Estimación Automática de la Porosidad del Viñedo Mediante Visión Artificial. 2016. Available online: https://consultas2.oepm.es/pdf/ES/0000/000/02/55/09/ES-2550903_B1.pdf (accessed on 23 March 2026).
10. Smart, R.; Robinson, M. *Sunlight into Wine: A Handbook for Winegrape Canopy Management*; Winetitles: Broadview, Australia, 1991; pp. viii+—88.
11. Tardáguila Laso, M.J.; Diago Santamaría, M.P.; Millán Prior, B.; Cubero García, S.; Aleixos Borrás, M.N.; Prats Montalbán, J.M. Patente de Invención con examen previo B2: Procedimiento Automático para Determinar la Compacidad de un racimo de uva en Modo Continuo, sobre una Cinta Transportadora sita en Bodega. 2015. Available online: [https://patents.google.com/patent/ES2523390B2/es?q=\(compacidad+de+racimo\)&inventor=tard%C3%A1guila&language=SPANISH](https://patents.google.com/patent/ES2523390B2/es?q=(compacidad+de+racimo)&inventor=tard%C3%A1guila&language=SPANISH) (accessed on 23 March 2026).
12. Cubero, S.; Diago, M.; Blasco, J.; Tardaguila, J.; Prats-Montalbán, J.; Ibáñez, J.; Tello, J.; Aleixos, N. A new method for assessment of bunch compactness using automated image analysis. *Aust. J. Grape Wine Res.* **2015**, *21*, 101–109. [CrossRef]
13. Su, S.; Chen, R.; Fang, X.; Zhu, Y.; Zhang, T.; Xu, Z. A novel lightweight grape detection method. *Agriculture* **2022**, *12*, 1364. [CrossRef]
14. Mohimont, L.; Alin, F.; Rondeau, M.; Gaveau, N.; Steffanel, L.A. Computer Vision and Deep Learning for Precision Viticulture. *Agronomy* **2022**, *12*, 2463. [CrossRef]
15. Casado-García, A.; Heras, J.; Milella, A.; Marani, R. Semi-supervised deep learning and low-cost cameras for the semantic segmentation of natural images in viticulture. *Precis. Agric.* **2022**, *23*, 2001–2026. [CrossRef]
16. Palacios, F.; Diago, M.P.; Melo-Pinto, P.; Tardaguila, J. Early yield prediction in different grapevine varieties using computer vision and machine learning. *Precis. Agric.* **2023**, *24*, 407–435. [CrossRef]
17. Huang, Y.; Qian, Y.; Wei, H.; Lu, Y.; Ling, B.; Qin, Y. A survey of deep learning-based object detection methods in crop counting. *Comput. Electron. Agric.* **2023**, *215*, 108425. [CrossRef]
18. Aguiar, A.S.; Monteiro, N.N.; dos Santos, F.N.; Pires, E.J.S.; Silva, D.; Sousa, A.J.; Boaventura-Cunha, J. Bringing semantics to the vineyard: An approach on deep learning-based vine trunk detection. *Agriculture* **2021**, *11*, 131. [CrossRef]
19. Pacioni, E.; Abengózar, E.; Macías, M.M.; García-Orellana, C.J.; Gallardo, R.; Velasco, H.M.G. Towards Intelligent Pruning of Vineyards by Direct Detection of Cutting Areas. *Agriculture* **2025**, *15*, 1154. [CrossRef]
20. García-Navarrete, O.L.; Correa-Guimaraes, A.; Navas-Gracia, L.M. Application of convolutional neural networks in weed detection and identification: A systematic review. *Agriculture* **2024**, *14*, 568. [CrossRef]
21. Albahar, M. A survey on deep learning and its impact on agriculture: Challenges and opportunities. *Agriculture* **2023**, *13*, 540. [CrossRef]
22. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
23. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
24. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*; Neural Information Processing Systems Foundation, Inc.: San Diego, CA, USA, 2012.
25. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
27. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
28. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
29. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
30. Redmon, 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
31. Jocher, G.; Chaurasia, A.; Qiu, J. YOLO by Ultralytics. GitHub. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 1 January 2023).

32. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
33. Shen, L.; Su, J.; He, R.; Song, L.; Huang, R.; Fang, Y.; Song, Y.; Su, B. Real-time tracking and counting of grape bunches in the field based on channel pruning with YOLOv5s. *Comput. Electron. Agric.* **2023**, *206*, 107662. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.