Article

Not peer-reviewed version

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

Valeriano Fuentes Méndez , Lourdes Lleó , Pilar Barreiro Elorza * , Abraham Tamargo-Vinces , Wilson Valente Da Costa Neto , Adolfo Moya Gonzalez , Pablo Guillén , Pilar Baeza

Posted Date: 27 March 2026

doi: 10.20944/preprints202603.2193.v1

Keywords: deep learning; computer vision; object detection

 doi:10.20944/preprints202603.2193.v1

Article

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

Valeriano Fuentes Méndez [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]

Technical University of Madrid

[1] Department Applied Mathematics, UPM
[2] LPF_TAGRALIA, Dept. Agroforestry engineering, UPM
[3] TAPAS, Dept. Plant Production, UPM
* Correspondence: pilar.barreiro@upm.es

Abstract

This study addresses the challenge of detecting white grape clusters (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 with occlusion issues. In this work, over 100 field RGB images were collected at La Bergonza (Toledo, Spain) and expanded through data augmentation, with various preprocessing strategies tested to enhance cluster visibility. Convolutional Neural Network (CNN) architectures were compared, highlighting YOLOv8 as superior to Mask R-CNN in both accuracy and efficiency. YOLOv8, trained for up to 100 epochs on equalized and augmented datasets, achieved outstanding performance: 84.9% precision, 72.6% recall, and mAP@0.5 of 83%, far surpassing Mask R-CNN (17% precision, 26% recall). The model successfully detected partially hidden clusters, including those invisible to human experts, better than previous studies that required controlled backgrounds or artificial lighting. Results confirm that combining RGB equalization with data augmentation optimizes detection. These findings underscore the potential of deep learning and low-cost RGB imaging systems to enable automated, scalable solutions for yield estimation and canopy analysis. In conclusion, YOLOv8 emerges as a promising tool for accurate grape bunch detection under field conditions, overcoming previous limitations.

Keywords: deep learning; computer vision; object detection

1. Introduction

The assessment of grape yield and quality in vineyard plots by experts is a hot topic in precision viticulture, with high human resource demand and low reliance when using a low number of reference plots. Among grape production quality, grape bunch compactness is a major issue. Thus, Tello & Ibañez (2014)[1] present over three seasons a detailed study of 24 morpho-agronomic variables related to compactness in different grape varieties, providing a detailed understanding of the influencing factors.

Following this research, Tello & Ibañez (2018) [2] review 19 morphological indices for assessing grape cluster compactness, made by a panel of experts which considers factors such as berry mobility and number of pedicels. The most relevant indices in relation to compactness were CI-12 = Cluster weight (g) / (Cluster length)², this was the one with the highest correlation and stands out for its simplicity; CI-18 which combines six variables: cluster weight, number of berries per cluster, number of seeds per berry, cluster length, length of the first branch and pedicel length; CI-19 also combines six variables very similar to CI-18 substituting pedicel length by the number of branches of the rachis. Furthermore, the importance of compactness in relation to grape and wine quality and susceptibility

 doi:10.20944/preprints202603.2193.v1

to fungal diseases is discussed, indicating that the development of image-based applications can be of great help in assessing compactness at ripeness level.

Working on image analysis, Herrero-Langreo et al. (2010)[3] proposed and validated a fast and simple method to classify RGB images of vineyards into six classes (clusters, green leaves, yellow leaves, shoots, trunk and canopy porosity) using the minimum Mahalanobis distance to assign the pixels into reference classes. This approach facilitates the accurate identification of different vineyard components, crucial for efficient crop monitoring and management.

Moreover, Correa et al. (2011)[4], working on 20 vineyard images, compared five Clustering Techniques: FCM, PCM, FPCM, RFPCM and FCM-GK; considering different color spaces (HSV, HSI, CMYK, Lab*, XYZ). The authors evaluated the percentage of well-classified pixels, using also images with added noise, and different resolutions and processing times. They concluded that the reduction of image resolution does not affect the performance of the techniques but significantly improves processing time. However, they found that RFPCM and PCM are not recommended due to the generation of overlapping clusters, which can lead to errors in classification.

In addition, Correa et al. (2012)[5] investigated the use of the Gustafson Kessel FCM algorithm to generate class centroids as seeds for K-means, improving the segmentation and classification of vineyard images. They compared this approach with FCM-GK, seedless K-Means and seeded K-Means, using the Lab* colour space. The results showed that the use of centroids generated by Gustafson Kessel FCM significantly improves segmentation accuracy.

Also, Diago et al. (2012)[6] implemented a supervised classifier based on Mahalanobis distance to characterize the vine canopy and to assess leaf area and yield using RGB images. The methodology automatically processes images and calculates areas corresponding to seven different classes (grapes, wood, background and four-leaf classes of different ages). This research evaluated 70 images of vines in La Rioja, Spain, the classification accounted for 92% and 98% well classifies items for leaves and bunches respectively: R^2 values of 0.81 for leaf area and 0.73 for production yield. The level of defoliation in the vineyard prior to image acquisition influenced the effectiveness of the yield estimation models; the higher the defoliation level, the higher the R2 value of the estimates.

The study by Íñiguez et al. (2021)[7] analyses the estimation of vineyard yield based on RGB images obtained from plots with different levels of leaf removal (none, partial, and total), which affects the visibility (occlusion) of the clusters. A white background was used for image capture, and then the components (clusters, leaves, branches, etc.) were segmented using Mahalanobis distance based on RGB and HSV (Hue and Saturation) values. Indices such as the percentage of occlusion, porosity, and exposure of clusters and leaves were calculated. The results show that the greater the occlusion, the worse the fit of the linear regression models between cluster pixels and actual production, with R^2 values dropping to 0.11 without defoliation and reaching 0.87 with total defoliation.

Iñiguez et al. (2024)[8] studies the development and application of different deep learning algorithms using YOLOV8 architecture on RGB images of vineyards in the field, considering a white background. The objective was to estimate the number of clusters and productivity in vineyards, considering different levels of cluster occlusion. Four models were developed from four calibration and validation sets, corresponding to different levels of leaf removal: raw image, partial leaf removal, and complete leaf removal. It was found that as the level of leaf removal increased, the model statistics also improved. A very high correlation was shown between the number of clusters predicted and observed by the expert in complete leaf removal, with worse results when the level of occlusion was high. This indicates the challenge that still exists in estimating the number of clusters and production when cluster occlusion levels are high.

Technology transfer towards the industrial sector has also been covered by several patents. Thus, Patent ES2550903B1[9] describes a process for automated estimation of vineyard porosity using machine vision. The process starts with the capture of an RGB image of a vine in the field using a digital camera, ensuring that other vines do not interfere in the quantification of the voids. The selected image is processed to obtain porosity information by segmenting the different elements of

 doi:10.20944/preprints202603.2193.v1

the image using the Mahalanobis distance. Pixels are then classified directly in the image as wood, leaves, clusters and backgroud. The porosity of the vineyard is estimated from the processed image, calculating the percentage of gaps as the ratio between the number of pixels labelled as background and the total number of pixels. A high coefficient of determination ($R^2 = 0.932$) is obtained between the estimation of gaps in the image and the conventional Point Quadrat system[10]. This procedure allows an accurate and efficient assessment of vineyard porosity, ensuring an optimal balance between leaf development and gaps to improve sun exposure and reduce fungal infections.

Another relevant patent, ES 2523390B2[11] describes a method for determining the compactness of grape clusters using RGB images in the winery. It uses a conveyor belt with varied backgrounds to better segment the images according to grape type. The process includes several stages: image segmentation into background, berry and rachis identification by means of supervised Bayesian classification (with prior training and obtaining probability functions); classification bunch voids using the watershed strategy; extraction of color and morphological features of the objects. A PLS model was established on a calibration set where the independent variables (X) consisted of the morphological and color variables, while the dependent variable (Y) corresponds to the averages compactness as evaluated by experts.

In the article by Cubero et al. (2015)[12], the PLS model showed 85.3% accuracy in cluster classification, highlighting the relevance of variables related to berry density and cluster shape for bunch compactness assessment.

Predicting grape yields accurately and quickly in plots has been the subject of a great deal of research in artificial intelligence (AI). Several methodologies have been used, such as YOLO and Faster R-CNN, SSD, and RetinaNet[13]. However, they reported a problem with the location of the bounding boxes (identifying the bunches).

Mohimont et al. (2022)[14] also introduce the use of computer vision techniques applied to precision viticulture, focusing particularly on the analysis of RGB images taken in the field. It explores supervised classification methods that enable the identification and segmentation of key elements such as grape clusters and leaves, facilitating the estimation of yield and leaf area. These techniques include the use of convolutional neural networks (CNNs), which are notable for their ability to extract complex visual patterns, and deep learning algorithms that improve accuracy in variable agricultural environments. The study also reviews approaches based on object detection and semantic segmentation, which allows tasks such as fruit counting and monitoring the vegetative state of the vine to be automated, all without the need for specialized sensors or controlled conditions.

Research by A. Casado-Garcia et al. (2022)[15] also examined deep learning for semantic segmentation (DeepLabV3+ with ResNext50, Manet with EfficientNetB3, among others) infield vineyard images using affordable RGB-D cameras. Five classes of interest were considered, one of them being the bunches of white grapes. The camera was mounted on a tractor and acquired infield images.

The study by Palacios et al. (2023)[16] proposes a method for estimating vineyard yield using nighttime artificial vision with LEDs and machine learning based on the SegNet DL architecture in CNN. The methodology includes segmentation of clusters and berries, creation of specific and global models, and calculation of production considering various parameters such as number of berries, pixels in leaves and branches, and average weight of berries. The SegNet model was trained in two phases with images of clusters and background to improve accuracy in berry segmentation. The paper follows a sequence consisting of pre-processing, image normalization, background removal, denoising, image feature extraction, preliminary raw result and post-processing to make it interpretable.

The review by Huang et al. (2023)[17], with over 130 references, classifies several problems when developing deep learning models for crop objects identification. Issues are classified into data scarcity, multi-scale detection and counting difficulty, together with severe occlusion in complex scenes. Data augmentation and transfer learning are the most frequently used methods to alleviate data scarcity. On the other hand, compared with large objects, small objects are more prone to false

 doi:10.20944/preprints202603.2193.v1

detection with adaptive multi-scale hybrid windows as a means to solve this bias. Although object detection and counting technology have made significant progress in recent years, occlusion is still one of the most critical challenges for detection and counting due to the complexity of the field scene. Several procedures to solve this issue include data augmentation, and data attention that focuses on local features of the image.

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

2. Materials and Methods

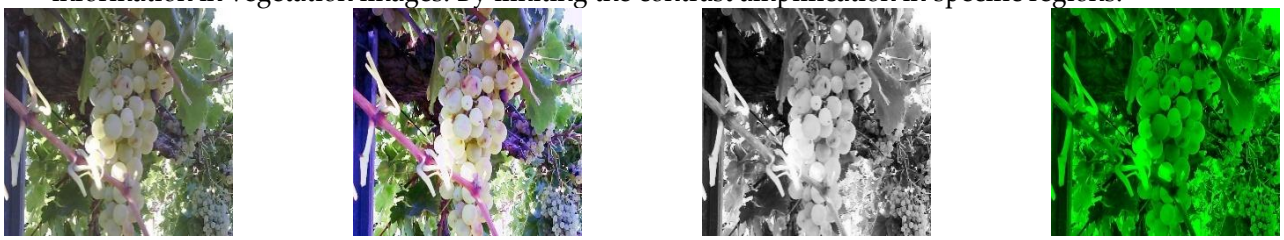
1. Description of the vineyard

The vineyard plot La Bergonza is located at 40.1534278N Latitude, -4.2207311 Longitude, (Toledo, Spain) and it belongs to the González-Byass winery, The vineyard, established in 2018 with the cv Airén (3.3m x 2m), is dedicated to high-intensity wine grape production, with yields between 30 and 40 t/ha. The single curtain training system shows a very dense grape-cluster area, with compact bunches but porous canopy development; considering a vine with a vine spacing of 2m along the row, the expected vine productivity is very high: 19.8-26.5 kg/ vine, with 77-103 bunches of grapes weighing a median of 256g/bunch an average Soluble Solids Content of grapes of 15.8+4.0 ºBrix.

2. Image pre-processing

A total of 326 images were captured, 177 for the training set, and 59 for the validation set. Different equalizations were used to train different detection models to evaluate the impact of each technique on training quality and model accuracy. Several histogram equalization techniques were applied to improve the visual quality in the detection of grape clusters. Four equalizations were implemented (Figure 1):

- Global RGB Channel Histogram Equalization: Global equalization was applied to the three-color channels (red, green and blue) of the images, which helped to improve the contrast in the color spectrum, making the grape clusters more visible in a complex environment where background colors may be similar.
- Global Gray Channel Histogram Equalization: this technique was applied to the grayscale channel of the images to improve the overall contrast, especially in those captured under unfavourable lighting conditions. Global equalization allows intensity levels to be more evenly distributed, making it easier for the model to better distinguish the clusters from the background.
- CLAHE (Contrast Limited Adaptive Histogram Equalization) of the Green Channel: CLAHE was applied specifically to the green channel, as this channel contains much of the relevant information in vegetation images. By limiting the contrast amplification in specific regions.



A B c d

Figure 1. Example of the equalization processes: a) original, b) RGB equalization, c) gray level image plus CLAHE, and d) equalization of RGB channel.

3. Methodology using CNN

Computer vision is a field of deep machine learning that allows the automatic identification of images. Although some basic models of images can be classified with the multilayer perceptron[18], it is with the use of convolutional networks[19] that the classification of images can be addressed in

 doi:10.20944/preprints202603.2193.v1

a general way. Relevant milestones in the development of convolutional networks are the AlexNet architectures[20], VGGNet[21], and finally ResNet, the residual networks[22].

The Region-based Convolutional Network (RCNN) method[23] first fine-tunes a convolutional network using a logarithmic loss function, then fine-tunes the features obtained in the convolutional network with SVMs, which act as object detectors that finally allow a regression of the bounding boxes. The Fast R-CNN architecture[24] improves the performance and accuracy of RCNN by a single-stage training using a multi-task loss.

Fast R-CNN starts from a complete image and a set of object proposals; the image is processed with several convolution and max-pooling layers to obtain a feature map. For each object proposal a region-of-interest (RoI) pooling layer extracts a fixed-length feature vector from the feature maps. The feature vector with a fully connected layer fits two branches to the bounding box and the class in which the object is classified. A final evolution is Mask R-CNN[25] where, in addition to the regression of a box containing the object and the classification of the object, a pixel map with the mask containing the identified object is added.

YOLO[26] is a network that allows identifying boxes containing the objects existing in a 2D scene together with the probability of belonging to a class. Different versions of this architecture have appeared up to version 8 which has been used in this work[27]. YOLOv8 predicts the containing boxes and the class probability without the need for a separate region proposal network. YOLOv8 has a center-based and anchorless approach to detect objects. It implements Pseudo-ensemble and Pseudo-supervision which involves training multiple models to generate a more diverse set of predictions, which improves the final accuracy and robustness.

The architecture used in YOLOv8 is the CSPDarknet53 backbone network (Figure 2), similar to GoogleNet[28], to extract image features, neck, and head. The most important changes compared to previous versions of YOLO are: 1) The CSP layer is replaced by the C2f module; 6x6 convolutions are changed to 3x3 ones; 1x1 convolutions are replaced by 3x3 convolutions at the bottleneck; The 10th and 14th convolutions are removed; A head decoupler is used, and the objectivity branch is removed.

The convolution blocks are made up of Conv2d, BatchNorm2d and SiLU activation. The bottlenecks are like residual blocks and are made up of two convolution blocks and a shortcut. The C2f module consists of a convolution block those branches to two bottlenecks or a direct step to the concatenation block. The SPPF (Spatial Pyramid Pooling Fast) block is made up of a convolution, three maxpooling layers and a concatenation ending in a convolution.

The backbone network is made up of 10 blocks (Figure 2), alternating Conv and C2f blocks and ending in an SPPF block. Thus, an RGB image of dimension 640x640 enters the backbone network and 512 maps of dimension 20x20 come out. The backbone has three outputs to the bottleneck, one from the final SPPF layer with dimension 20x20 and two others from the 5th block with dimension 80x80 and from the 7th block with dimension 40x40. The bottleneck has vertical connections with up sampling steps changing the dimension from 20 to 40 and 80. In turn it connects laterally with the header that also has vertical connections, using Conv blocks to decreases the dimension from 80 to 40 and 20 and vertically to three Detect blocks, in dimensions 20, 40 and 80. Each Detect block branches into 2 lines with a pair of Conv with filter 3 ending in a Conv2d with filter 1, the first line calculates the container box loss and the second one the class loss; YOLOv8 also has a segmentation module called YOLOv8-Seg that includes two semantic segmentation heads.

 doi:10.20944/preprints202603.2193.v1

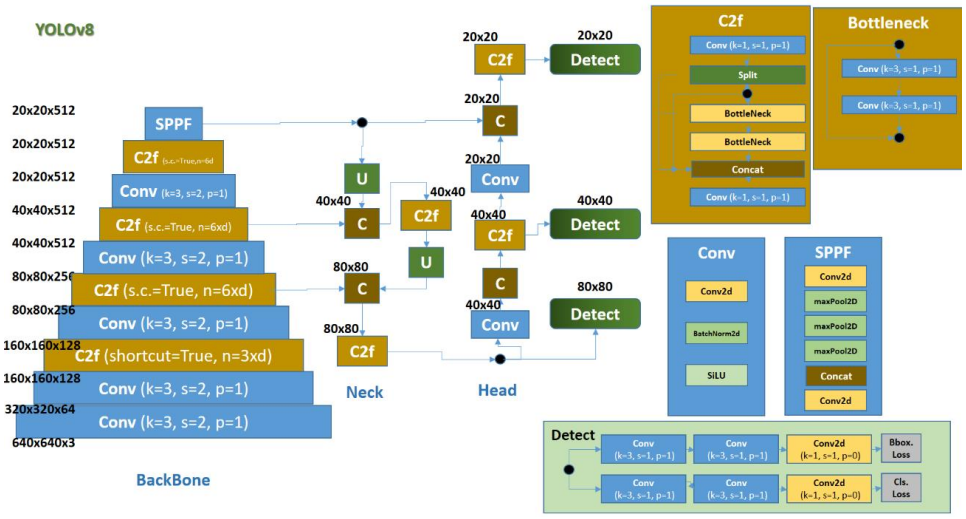


Figure 2. Overview of the YOLOv8 architecture. On the left, the convolutional backbone network is based on Conv and C2f layers. The backbone network starts with three RGB channels of dimension 640 and ends with 512 channels of dimension 20. At two intermediate points and the end of the backbone network, there is an output to the neck and detector head. Detection allows for the extraction of the containing boxes, classes, and masks. U is up sampling process to increase the map size; C is a process to connect the two maps inputs into one.

The training process (Figure 3) starts with the loading of the files with the images and the annotations. The images are optionally equalized, and the result forms a tensor of dimensions (3,w,h) where (w,h) is the width and height for the 3 RGB color channels. The annotations allow obtaining the target with the class of the objects and the container box and optionally, the mask, for the mask-RCNN or Yolov8 model with segmentation.

The next step is data augmentation, all of which allows obtaining a dataset that is shuffled into training, testing and validation. The dataset is managed with a dataloader that will be used for training and validation. The data augmentation allows increasing the number of elements to train the model based on random noise, cropping, flipping and rotation processes.

In this paper YoloV8 procedure was used considering up to 100 train epochs. Several metrics are considered: model score threshold, true Positives (TP), false Positives (FP), false Negatives (FN), total Predict Box (PB), total Real Box (RB), total Real Box (train dataset), Precision (TP/(TP + FP)), Recall (TP/(TP+FN)), and maximum score in predicted box.

 doi:10.20944/preprints202603.2193.v1

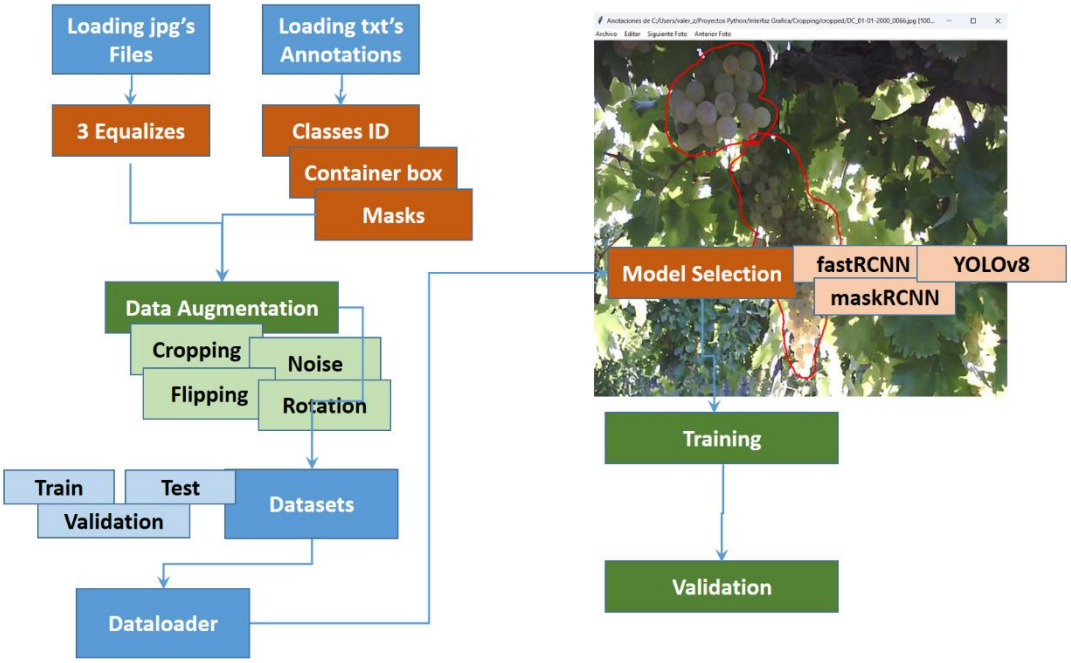


Figure 3. Training process. The top right shows how the dataset information is saved: the image files, and the annotated text files. Annotations can be the vertices containing the boxes or the mask polygon. The data augmentation process (cropping, noise, flipping, and rotation) increases the number of elements in the dataset to its final value. This augmented dataset is divided into Train and Test for the training process, and Validation for validation, managed from a Dataloader. The training process can be performed with any of the implemented models (fastRCNN, maskRCNN, or YOLOv8).

The number of epochs influences the improvement in model fit. In each epoch, the network configuration is updated by checking the predictions with ground-truth. Specifically, in the maskRCNN tests, 100 epochs were run, and with YOLOv8, 50 epochs were run first, then up to 100 epochs in total.

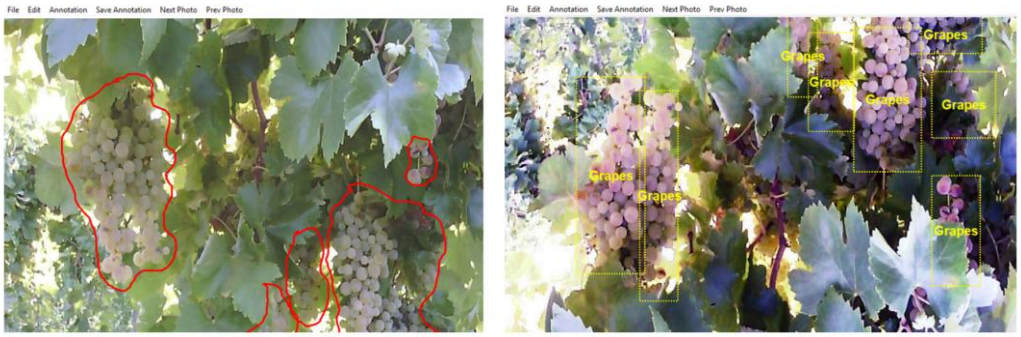


Figure 4. On the left is an image showing the annotations used in maskRCNN. These are constructed with a closed polygonal line that becomes the mask required by maskRCNN. On the right are the annotations used in YOLOv8, based on bounding boxes.

1. 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.

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

If IoU=1 then the prediction is perfect, while an 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

 doi:10.20944/preprints202603.2193.v1

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 doesn't exist in the ground-truth. In this paper a range between 0.35 and 0.65 for 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 to 6).

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

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

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

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

- **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

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

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

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (5)$$

- **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).

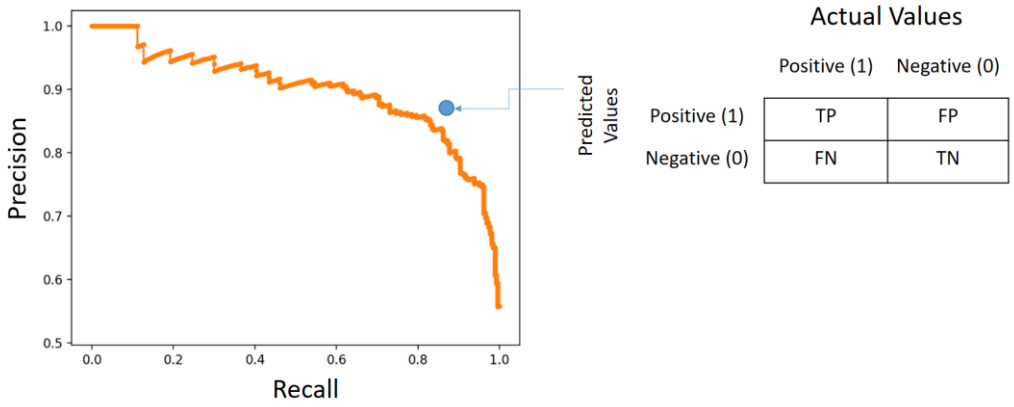


Figure 5. By changing the IoU threshold value, all the containing boxes are evaluated, and based on the resulting value (TP, FP, FN, and TN), the confusion matrix is obtained with the count of all of them (right). Using the data from the confusion matrix, the precision and recall values are obtained by drawing the PR-Recall graph, on the left in orange. The integral under the PR-Recall graph allows 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.

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

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

 doi:10.20944/preprints202603.2193.v1

- **mAP@0.5:** average of the APs with an IoU threshold of 0.5. If the IoU ≥ 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

Table 1 and Table 2 summarize the results on grape-bunches identification using the MaskRCNN procedure considering: 100 train epochs, 177 Images in training set, and 59 images for the testing set. Several model are trained for varied: IoU and model score threshold, leading to varied true Positives (TP), false Positives (FP), false Negatives (FN), total Predict Box (PB), total Real Box (RB), total Real Box (train dataset), Precision (TP/(TP + FP)), Recall (TP/(TP+FN)), and maximum score in predicted box. The best results are found for score threshold of 0.65 leading to a maximum score in predicted grape-bunch box of 0.998 and minimizing both false positives and false negatives (197 and 111 respectively). In general terms, an excessive number of grape-bunches identification is found (161%), that is 237 labelled regions compared to 147 actual grape-bunches (table 3, score threshold 0.65). In this table, it can be observed that very poor results were attained: score threshold 0.35; true Positives 67; false Positives 653; false Negatives 101; total Predict Box 6; total Real Box 175; total Real Box (train dataset) 720; Precision 0.093; and Recall 0.399.

Table 1. Results for the maskRCNN procedure considering 100 train epoches; 177 Images in training set; and 59 images the testing set, for varying IoT threshold, leading to true Positives (TP), false Positives (FP), false Negatives (FN), Precision (TP/(TP + FP)), Recall (TP/(TP+FN)).

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 for the maskRCNN procedure considering 100 train epochs: model score threshold, total Predict Box (PB), total Real Box (validation dataset), total Real Box (train 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

 doi:10.20944/preprints202603.2193.v1

0.65	0.998	237	147	161%	520

Figure 6 shows an example of the identification of grape bunches using various maskCNN models. It can be observed that some of the models show improved performance when identifying grape-bunches that are not clear for the naked eye.

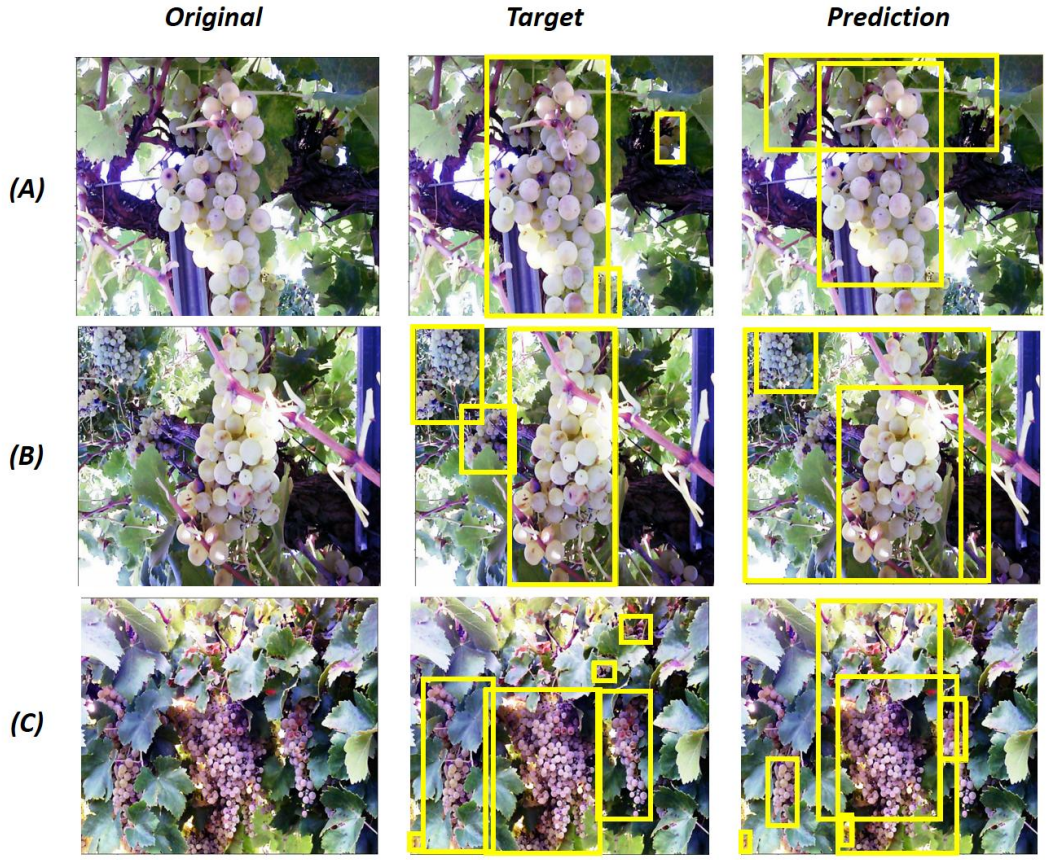


Figure 6. three examples grape bunches (A,B,C) identified by means of several maskCNN models; in several grape bunches that are not clear for the naked eye are labelled.

Table 3 shows the ability of the YOLOv8 model to detect grape clusters by applying different image equalization techniques and 50 and 100 epochs training. RGB equalization and the use of a combined filter achieved the best results in terms of both mAP. The RGB equalization stood out for providing higher accuracy, while the combined filter showed the best balance between accuracy and recall, achieving a superior fitness value.

Table 3. Results for the YoloV8 procedure considering 50 train epochs (first row) and 100 train epochs (second row); 495 Images in training set; and 124 images for the validation set.

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

 doi:10.20944/preprints202603.2193.v1

	0.849	0.726	0.839	0.465	0.503

Table 3 shows several metrics: fitness, recall, map50 and mapAP50-95 for YOLOv8 models. It can be concluded that the best model corresponds to the augmented dataset applied to RGB equalized images. The four models have precision values above 75%, while recall values being above 61%.

Table 4. Calculated accuracy from confusion matrix data in different models.

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%

In object detection, accuracy is not usually a useful or reported metric, since the number of true negatives (TN) is huge and undefined. However, by omitting the TN value, an accuracy value can be obtained from the confusion matrix, as shown in Table 4. Highest accuracy is obtained RGB equalization with augmented dataset.

4. Discussion

This paper shows that the results obtained using YOLOv8 are better than those obtained using Mask R-CNN. This is because YOLOv8 is a newer architecture that incorporates more sophisticated loss functions, better integrated batch normalization and dropout, and lighter and more efficient backbones

Palacios et al. (2023)[16] use SegNET to detect berries in grape bunches obtained with active illumination in nocturne acquisition, while in our case Yolov8 is used to directly detect grape bunches with daylight and passive illumination.

Iñiguez et al. (2021)[7] analysed the influence of different levels of leaf occlusion on the performance of models for bunch detection and yield estimation. Different linear regressions between pixels detected by computer vision and yield were analysed. The main conclusion being that the level of occlusion on bunches affects the machine's performance in yield assessment.

Iñiguez et al. (2024)[8] uses Yolov4 in day light to the detect red grape bunches using white background to improve segmentation. In our case, white grapes with no background manipulation is used. Our results excel in terms of precision, with similar values for recall. Thus, in our case precision reaches 0.84 compared to 0.68 for Iñiguez; our recall values reached 0.73 compared to 0.74.

Casado-García et al. (2022)[15] tested various neural networks on 85 labelled images. Results showed that DeepLabV3+ with ResNext50 achieved 84.78% accuracy, while Manet with EfficientNetB3 excelled at identifying grape bunches with 85.69%. Semi-supervised learning improved accuracy by around 5.6% to 6%, demonstrating its benefit in agricultural imaging. These results present better performance than current research where the achieved accuracy accounted to 61%. The experiments were developed considering white grapes and infield and natural illumination conditions.

Su et al. (2022)[13] proposes a method for detecting grape bunches by means of bounding boxes. The method is tested in the field in sunny and cloudy conditions, with different degrees of ripeness and coloration of the grapes, as well as partial coverage of grape clusters by the leaves. The method is constituted by three parts: a backbone network to enhance feature extraction, then a Bi-directional Path Aggregation Network to fuse different scale feature maps to improve feature information, and the last part the Relocation of Non-Maximum Suppression R-NMS algorithm to improve the accuracy of the location of bounding boxes. The method is based on YOLOv4. Considering IoU of 0.5, and 150

 doi:10.20944/preprints202603.2193.v1

training epochs, they obtained 87,7% mAP, 88,6% precision, 78,3% recall and 83,1% of F1 score. These results are slightly better than ours.

In the review by Huang et al. (2023)[29], Shen et al. (2023) shows for grape detection an average counting accuracy of 84.9%, and a correlation coefficient with manual counting of 0.9905; features like those of Casado-García et al. (2022)[15].

5. Conclusions

The YOLOv8 architecture demonstrated superior performance compared to Mask R-CNN, achieving higher precision (84.9%) and recall (72.6%) under field conditions. This confirms its robustness for detecting grape clusters in complex vineyard environments without requiring artificial backgrounds or lighting.

Applying RGB histogram equalization combined with data augmentation significantly improved model fitness and detection metrics. This preprocessing strategy optimized bunch visibility in dense canopies, reducing occlusion-related errors and enhancing overall detection reliability.

The integration of deep learning models with low-cost RGB imaging systems offers a practical and scalable solution for automated yield estimation and canopy analysis. This approach reduces dependency on manual labour and expert assessments, paving the way for widespread adoption in commercial vineyards.

Automated grape bunch detection supports precision viticulture practices that improve resource efficiency and crop monitoring. By enabling accurate yield prediction and canopy management, these technologies contribute to sustainable and resilient agricultural systems in the face of climate variability.

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 maskRCNN and YOLOv8 processes.

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

Conflicts of Interest: 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
RCNN	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

 doi:10.20944/preprints202603.2193.v1

1. Tello, J., & Ibáñez, J. (2014). Evaluation of indexes for the quantitative and objective estimation of grapevine bunch compactness. Vitis-Geilweilerhof, 53, 9–16.
2. Tello, J., & Ibáñez, J. (2018). What do we know about grapevine bunch compactness? A state-of-the-art review. Australian Journal of Grape and Wine Research. 24(1), 6-23.
3. Herrero-Langreo, A., Barreiro, P., Diago, M. P., Baluja, J., Ochagavia, H., & Tardaguila, J. (2010). Pixel classification through Mahalanobis distance for identification of grapevine canopy elements on RGB images. International Association for Spectral Imaging (IASIM-10)
4. Correa, C., Valero, C., Barreiro, P., Diago, M. P., & Tardáguila, J. (2011). A comparison of fuzzy clustering algorithms applied to feature extraction on vineyard. Inteligencia Artificial: Revista Iberoamericana de Inteligencia Artificial.
5. Correa, C., Valero, C., Barreiro, P., Diago, M. P., & Tardáguila, J. (2012). Feature extraction on vineyard by Gustafson Kessel FCM and K-means. In Proceedings of the Mediterranean Electrotechnical Conference (MELECON).
6. Diago, M. P., Correa, C., Millán, B., Barreiro, P., Valero, C., & Tardaguila, J. (2012). Grapevine yield and leaf area estimation using supervised classification methodology on RGB images taken under field conditions. Sensors, 12, 16988–17006.
7. Íñiguez, R., Palacios, F., Barrio, I., Hernández, I., Gutiérrez, S., & Tardaguila, J. (2021). Impact of leaf occlusions on yield assessment by computer vision in commercial vineyards. Agronomy, 11, 1–13.
8. Íñiguez, R., Gutiérrez, S., Poblete-Echeverría, C., Hernández, I., Barrio, I., & Tardáguila, J. (2024). Deep learning modelling for non-invasive grape bunch detection under diverse occlusion conditions. Computers and Electronics in Agriculture, 226.
9. Tardáguila Laso, M. J. M. P. B., & D. S. M. P. (2016). Patente de invención B1: Procedimiento para la estimación automática de la porosidad del viñedo mediante visión artificial.
10. Smart, R., & Robinson, M. (1991). Sunlight into wine: a handbook for winegrape canopy management (pp. viii+-88).
11. Tardáguila Laso, M. J. M. P. B., & D. S. M. P. (2015b). 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.
12. Cubero, S., Diago, M. P., Blasco, J., Tardaguila, J., Prats-Montalbán, J. M., Ibáñez, J., Tello, J., & Aleixos, N. (2015). A new method for assessment of bunch compactness using automated image analysis. Australian Journal of Grape and Wine Research, 21, 101–109.
13. Su, S., Chen, R., Fang, X., Zhu, Y., Zhang, T., & Xu, Z. (2022). A novel lightweight grape detection method. Agriculture (Switzerland), 12. [**] Mohimont, L., Alin, F., Rondeau, M., Gaveau, N., & Steffenel, L. A. (2022). Computer vision and deep learning for precision viticulture. Agronomy.
14. Mohimont, L., Alin, F., Rondeau, M., Gaveau, N., & Steffenel, L. A. (2022). Computer Vision and Deep Learning for Precision Viticulture. In Agronomy (Vol. 12, Number 10). MDPI.
15. Casado-García, A., Heras, J., Milella, A., & Marani, R. (2022). Semi-supervised deep learning and low-cost cameras for the semantic segmentation of natural images in viticulture. Precision Agriculture, 23, 2001–2026.
16. Palacios, F., Diago, M. P., Melo-Pinto, P., & Tardaguila, J. (2023). Early yield prediction in different grapevine varieties using computer vision and machine learning. Precision Agriculture, 24, 407–435.
17. Huang, Y., Qian, Y., Wei, H., Lu, Y., Ling, B., & Qin, Y. (2023). A survey of deep learning-based object detection methods in crop counting. Computers and Electronics in Agriculture.
18. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533–536.
19. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4), 541–551.
20. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25.
21. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

 doi:10.20944/preprints202603.2193.v1

22. Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 580–587).
23. Girshick, R. (2015). Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1440–1448).
24. He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2961–2969).
25. He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning. Image Recognition, 7(4), 327–336.
26. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 779–788).
27. Jocher, G., Chaurasia, A., & Qiu, J. (2023, January 1). YOLO by Ultralytics. GitHub. https://github.com/ultralytics/ultralytics.
28. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1–9).
29. Shen, L., Su, J., He, R., Song, L., Huang, R., Fang, Y., Song, Y., & Su, B. (2023). Real-time tracking and counting of grape clusters in the field based on channel pruning with YOLOv5s. Computers and Electronics in Agriculture, 206.