COUNTING GRAPE BUNCHES USING DEEP LEARNING UNDER DIFFERENT FRUIT AND LEAF OCCLUSION CONDITIONS

Authors: Rubén ÍÑIGUEZ[1,2], Carlos POBLETE- ECHEVERRIA[1,2], Inés HERNÁNDEZ[1,2], Salvador GUTIÉRREZ[3], Ignacio BARRIO[1,2] and Javier TARDÁGUILA[1,2*]

[1]*Televitis Research Group, University of La Rioja, 26006 Logroño, Spain*
[2]*Institute of Grapevine and Wine Sciences (University of La Rioja, Consejo Superior de Investigaciones Científicas, Gobierno de La Rioja), 26007 Logroño, Spain*
[3]*Department of Computer Science and Artificial Intelligence (DECSAI), Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada (UGR), 18071 Granada, Spain*

*Corresponding author: *javier.tardaguila@unirioja.es*

Keywords: artificial intelligence; yield estimation; YOLO; precision viticulture; object detection.

Introduction

Yield estimation is very important for the wine industry since provides useful information for vineyard and winery management. The early yield estimation of the grapevine provides information to winegrowers in making management decisions to achieve a better quantity and quality of grapes. In general, yield forecasts are based on destructive sampling of bunches and manual counting of berries per bunch and bunches per vine. This traditional approach does not provide accurate estimations because the sample of the vineyard cannot represent all the variability that may be present in the plot. These techniques are time-consuming, expensive, and labour-intensive (Martin et al., 2003). The number of bunches per vine is the most important of the yield components, explaining 60% of average field yield variability, while the number of berries per bunch explains 30% and berry weight explains 10% (Laurent et al., 2021). In this regard, precision viticulture has brought new opportunities for yield monitoring and prediction, taking advantage of the new sensors, platforms, and modelling techniques.

Nowadays one of the most common and successful techniques for monitoring the amount of fruit in viticulture has been computer vision. Several applications and methods have been reported in the scientific literature (Mohimont et al., 2022). Computer vision systems have been used to estimate grapevine yield at different phenological stages, such as budbreak (Liu et al., 2017), flowering (Palacios et al., 2020), pea-size (Palacios et al., 2022), and harvest (Xin et al., 2020). The computer vision techniques used for bunch detection are mainly classified into three classes: i) colour-based thresholding and colour features (Hacking et al., 2020), ii) active contour segmentation (Xiong, 2018), and ii) pixels segmentation (Íñiguez et al., 2021). In general, computer vision has shown good results for bunch detection; however, the results of these techniques are highly influenced by image acquisition conditions such as background effects and light conditions and intrinsic conditions of the grape canopies such as bunch occlusion (Íñiguez et al. 2021). In this context, new artificial intelligence techniques can help us to solve these problems. Deep learning methods have proved to be very effective in object detection (Fuentes, 2017). This novel technique has shown promising results for bunch detection and counting in grapevines (Sozzi et al., 2022).

Research Objectives

The objective of this study was to analyse and quantify the effect of fruit and leaf occlusion on the performance of a deep learning algorithm (YOLOv4) used for automatic grape bunch counting under field conditions.

Material and methods

Experimental sites and layout

The experiment was conducted in 2020 in eleven commercial dry-farmed cv. Tempranillo (*Vitis vinifera* L.) vineyards located in Rioja wine appellation, Spain. All vineyards were spur-pruned and trained on a vertical shoot positioning trellis system with two pairs of movable wires. All vineyard plots were subject to similar standard cultural practices during the growing season: de-suckering, shoot positioning and shoot trimming. No defoliation was performed before image acquisition at harvest.

In each vineyard site, 25 single vines were randomly chosen before harvest. All vines were divided into two segments of 0.5 m and labelled accordingly. The vine canopy was successively defoliated: first by removing the first four main basal leaves (partial defoliation), and then the remaining main leaves and laterals (full defoliation). Images were taken in the vineyard for each individual segment before each defoliation step.

Image acquisition

The canopy images were taken with a conventional RGB camera (Canon EOS 5D Mark IV RGB, Canon Inc. Tokyo, Japan) with no artificial illumination (uncontrolled environment conditions). The camera was mounted on a tripod set pointed to the canopy, at 1.0 m from the row axis and 1.20 m aboveground. A white screen was placed behind the canopy to remove the influence of background vegetation. Images were saved in JPG format with the highest quality setting available in the camera. The full image size was 6720 × 4480 pixels.

Ground truth data

After image acquisition bunches were harvested and counted manually per each vine segment (actual number of bunches). A region of interest (ROI) was defined manually to analyse the canopy segment for defoliation level. The images inside of the ROI were manually labelled, selecting visible bunches with bounding boxes using LabelImg software (Tzutalin, 2015). Bunch class was used to label the dataset of this study using the YOLO label format.

Object detection algorithm

Object detection was modelled from the YOLOv4 architecture (Bochkovskiy et al., 2020), implemented with *darknet* (Redmon, 2013) and using a modified configuration from the original Bochkovskjy's published code (Bochkovskiy, 2020). The input of the network was 3×320×320 (channels, height, and width, respectively) with a batch size of 256 images. Regarding the training control parameters, momentum was statically set to 0.9, but the learning rate started with a value of 0.001 and was dynamically adjusted during the training with a decay value of 0.0005. The number of training iterations were capped up to 20000. To further increase the generalization capability of the model, data augmentation was performed from the original images. Prior to training, in an offline pipeline, random

colour and transformation parameters were applied to increase the training dataset: alterations in saturation and value channels (HSV colour space), image rotation, blurring and flipping. Training was deployed in a server with a 64 thread AMD Ryzen Threadripper 3970X 32-Core CPU, 32 GB of RAM and one nVidia GeForce RTX 3090 GPU. The full training process needed around 24 hours to complete.

Statistical analysis

To evaluate the performance of YOLOv4 for bunch detection, the following indicators were used: i) mean average precision (mAP), which summarizes the precision of the model; ii) the Intersection over Union (IoU) which evaluate the effectiveness in the overlap between the bounding boxes (labelled and predicted); iii) Precision that is the ratio between the number of correctly detected and the total number of objects detected; iv) Recall that is the ratio between the number of correctly detected and the number of all the bunches; and v) F1-score that is defined as the harmonic mean of precision and recall. Additionally, to evaluate the predictions of the model, the root mean squared error (RMSE) and the coefficient of determination (R^2) were used.

Results

The effect of fruit occlusion was evaluated in the manual method through the comparison between the actual number of bunches per segment (manual counting) and the number of bunches visible in the RGB images (Figure 1). This basal effect exits and represents an RMSE of 1.35 bunches (Table 1). When the leaf occlusion appears (partially defoliated and no defoliated) the error in the detection increases with an RMSE of 1.41 and 1.71, respectively. The linear regression between the manual and the visible number of bunches indicates a strong correlation. The R^2 reached for the three levels of occlusion were 0.86 for the full defoliated (no occlusion), 0.83 for the partial defoliated (medium occlusion) and 0.81 for no defoliated (high occlusion) (Table 1).

The performance of the deep learning algorithm (YOLOv4 architecture) was analysed by the comparison with visible bunches counted manually in each image. The results of the validation demonstrated that the YOLOv4 architecture achieved high values of performance with a mAP of 90.19 %, a Recall of 0.9, a F1 score of 0.87 and a Precision of 0.85 for the optimal conditions (fully exposed bunches) (Table 1). An example of the output of YOLOv4 is presented in Figure 1, which shows the original images for the three levels of leaf removal, the images labelled by an expert and the result of the prediction made by the deep learning algorithm. The linear regression between the visible number of bunches (labelled) versus the predicted number of bunches (fully defoliated canopies) confirms the good performance of YOLOv4 with an R^2 of 0.85 and an RMSE of 1.07. The accuracy was slightly lower when the model was tested in partially defoliated canopies with an R^2 of 0.79 and an RMSE of 1.22, however, the accuracy was drastically reduced when the model was tested in no defoliated canopies with an R^2 of 0.64 and an RMSE of 1.48. This reduction of accuracy indicates the effect of leaf occlusion since the model was not totally able to detect bunches partially visible. The lack of detection under these conditions can be attributed to the training process since the model was trained on images with fully visible bunches.

Conclusion

The deep learning algorithm YOLOv4 was able to detect the number of bunches with high precision under full leaf defoliation and partial defoliation conditions. The leaf and fruit occlusion negatively affects the performance of the model. The reduction of the accuracy is a clear effect of leaf and fruit occlusion since the model was not able to detect bunches partially visible. The lack of detection on no defoliated canopies can be attributed to the training process since the model was trained on images with fully visible bunches. Further studies are needed to test this principle, performing the training process with images in which partially occluded bunches appear to incorporate this feature in the detection algorithm.

Acknowledgements

Research founding FPI Grant 591/2021 by Universidad de La Rioja, Gobierno de La Rioja.

Literature Cited

Bochkovskiy, A., Wang, C. Y., Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

Bochkovskiy, A. (2020). Yolo v4 repository [source code]. https://github.com/AlexeyAB/darknet

Fuentes, A., Yoon, S., Kim, S.C., Park, E.D.S. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors, 17.

Hacking, C., Poona, N., Poblete-Echeverria, C. (2020). Vineyard yield estimation using 2-D proximal sensing: A multitemporal approach. OENO One, 54, 793–812

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(5), 1003.

Laurent, C., Oger, B., Taylor, J. A., Scholasch, T., Metay, A., Tisseyre, B. (2021). A review of the issues, methods and perspectives for yield estimation, prediction and forecasting in viticulture. European Journal of Agronomy, 130, 126339.

Liu, S., Cossell, S., Tang, J., Dunn, G., Whitty, M. (2017). A computer vision system for early stage grape yield estimation based on shoot detection. Computers and Electronics in Agriculture, 137, 88–101.

Luo, L., Tang, Y., Zou, X., Wang, C., Zhang, P., Feng, W. (2016). Robust grape cluster detection in a vineyard by combining the adaboost framework and multiple color components. Sensors, 16, 2098.

Martin, S., Dunstone, R., Dunn, G. (2003). How to forecast wine grape deliveries using grape forecaster excel workbook version 7; Department of Primary Industries: Adelaide, Australia.

Mohimont, L., Alin, F., Rondeau, M., Gaveau, N., Steffenel, L.A. (2022). Computer vision and deep learning for precision viticulture. Agronomy, 12, 2463.

Palacios, F., Bueno, G., Salido, J., Diago, M. P., Hernández, I., Tardaguila, J. (2020). Automated grapevine flower detection and quantification method based on computer vision and deep learning from on-the-go imaging using a mobile sensing platform under field conditions. Computers and Electronics in Agriculture, 178, 105796.

Palacios, F., Melo-Pinto, P., Diago, M. P., Tardáguila, J. (2022). Deep learning and computer vision for assessing the number of total berries and yield in commercial vineyards. Biosystems Engineering, 218, 175–188.

Redmon, J. (2013). Darknet: Open source neural networks in C.

Reis, M.J.C.S., Morais, R., Peres, E., Pereira, C., Contente, O., Soares, S., Valente, A., Baptista, J., Ferreira, P.J.S.G., Bulas Cruz, J. (2012) Automatic detection of bunches of grapes in natural environment from color images. Journal of Applied Logics, 10, 285–290.

Sozzi, M., Cantalamessa, S., Cogato, A., Kayad, A., Marinello, F. (2022). Automatic bunch detection in white grape varieties using YOLOv3, YOLOv4, and YOLOv5 deep learning algorithms. Agronomy, 12, 319.

Tzutalin, D. (2015). LabelImg. GitHub Repository, 6.

Xin, B., Liu, S., & Whitty, M. (2020). Three-dimensional reconstruction of Vitis vinifera L. cvs Pinot Noir and Merlot grape bunch frameworks using a restricted reconstruction grammar based on the stochastic L-system. Australian Journal of Grape and Wine Research, 26(3), 207–219.

Xiong, J., Liu, Z., Lin, R., Bu, R., He, Z., Yang, Z., Liang, C. (2018) Green grape detection and picking-point calculation in a night-time natural environment using a charge-coupled device (ccd) Vision Sensor with Artificial Illumination. Sensors, 18, 969.

Tables and Figures

Table 1. Results of the bunch estimation (R^2 and RMSE) for the comparison of the number of actual bunches counted in field and the number of visible bunches counted by an expert at the images. Results of performance of the YOLOv4 model for bunch detection with the metrics (mAP, Recall, Precision, IoU, F1, R^2, RMSE) for training and validation datasets and results of the object detection with the comparison of YOLOv4 prediction with visible number of bunches for the validation dataset.

Results of regression on the number of actual bunches against the number of visible bunches			
	Full defoliated	**Partial defoliated**	**No defoliated**
R^2	0.86	0.83	0.81
RMSE	1.35	1.41	1.71

Performance of the model					
	mAP	**Recall**	**Precision**	**IoU**	**F1**
Training	99.9	1.0	0.98	89.84	0.99
Validation	90.1	0.9	0.85	70.67	0.87

Results of regression on the number of visible bunches against the number of predicted bunches			
	Full defoliated	**Partial defoliated**	**No defoliated**
R^2	0.85	0.79	0.64
RMSE	1.07	1.22	1.48

R^2: determination coefficient; RMSE: root mean square error; mAP: mean average precision (%); IoU: intersection over union.

Figure 1. Comparison of the number of visible bunches in an original image, detected and labelled by an expert (blue bounding boxes) and predicted bunches using YOLOv4 (red bounding boxes) in full defoliation (a), partial defoliation (b) and no defoliation (c).