



Automated Detection of Grapes and Leaves in Viticulture with a YOLOv2 Neural Network

Andreza Santos*, Sandra Avila, Thiago T. Santos

Abstract

In this work, we modeled the problem of detection of fruit and leaves in viticulture for proximal applications as a supervised machine learning task. We created and manually labeled a database of images obtained at Guaspari Winery. In total, the database consists of 11.883 images of bunches of grapes and leaves. We trained a convolutional network with YOLOv2 architecture to locate and classify bunches of grapes and leaves. Quantitative tests have shown results for detection and classification with precision of 100%, recall of 74.2% and F1-Score up to 85.2% for the class "grape". Also, qualitative tests show that the model generalizes well when tested on photographs of other grape varieties. These results are promising and are moving towards the possibility of application in the field.

Key words:

Fruit detection, Viticulture, Deep Learning

Introduction

The detection and classification of fruits are essential components in automation applications in the field of precision agriculture. Due to fruit size, proximal sensing becomes more appropriate where image-based techniques and neural networks constitute the current state of the art [1]. As the goal is to be able to predict through images from a camera that is embedded in a mobile field system, we chose the YOLOv2 neural network [2] for the task because of its real-time performance on prediction.

Results and Discussion

We built a database from images obtained from Guaspari Winery. The samples are composed of bounding boxes containing grape clusters and leaves. Due to the limitation of the number of images, some of them went through the process of data augmentation.

The YOLOv2 neural network architecture presents 24 convolutional layers followed by two fully connected layers. The network produces a confidence index, which reflects the model's confidence that an object exists within a given bounding box and how accurate it is. As an output from the network, we get the predicted bounding boxes that present a confidence index greater than or equal to a threshold. We initialized the training with the weights of a YOLOv2 network pre-trained with the ImageNet database. A sequence of 90,000 forward/backward training operations was performed. The evaluation metrics used were precision, recall, and F1-Score for each class.

Figure 1 shows a detection result using the best parameters found. In Table 1, we can observe that from the threshold equal to 0.6 we have an increase in the number of false negatives, indicating that for some samples the network could not predict with threshold above the acceptable. This leads to a 1.6% F1-Score for a 0.9 threshold.

We can attribute the results obtained to the lack of diversity in the samples since they are obtained from well-behaved photographs. It is possible that by adding examples to the database that cover a greater range of defects, it can produce better results.

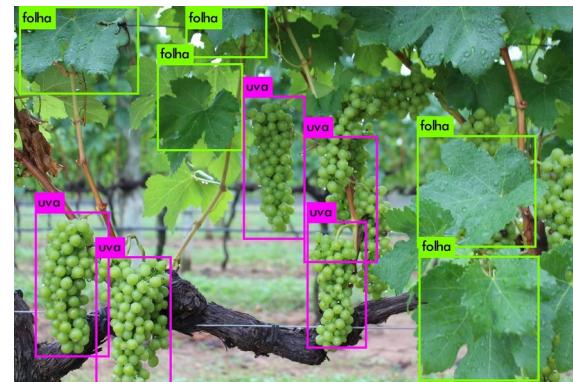


Figure 1. Detection results using a threshold equal to 0.6 and weights of interaction 90.000.

Table 1. Precision, Recall and F1-score for the class "grape".

Threshold	Precision	Recall	F1-Score	False Negative
0.5	1	74.2%	85.2%	33
0.6	1	73.2%	84.6%	34
0.7	1	65.4%	79.1%	44
0.8	1	47.2%	64.2%	67
0.9	1	0.8%	1.6%	126

Conclusions

In general, the YOLOv2 neural network presented promising results for the introduced detection problem. As future work, we intend to improve these results and test other alternatives from the same architectural family as YOLOv3, Fast-YOLO and another family such as Mask R-CNN.

Acknowledgement

The authors gratefully thank Embrapa, PIBIC/CNPq (#161165/2017-6), NVIDIA's GPU Grant Program, and Microsoft.

¹ Sa, I., et al. "DeepFruits: A fruit detection system using deep neural networks. Sensors, vol. 16, n. 8, p. 1222. 2016.

² Redmon, J.; Farhadi, A. "YOLO9000: better, faster, stronger." In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. p. 6517-6525.