Computers and Electronics in Agriculture 170 (2020) 105247

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association

Thiago T. Santos[a,*], Leonardo L. de Souza[b], Andreza A. dos Santos[b], Sandra Avila[b]

[a] Embrapa Agricultural Informatics, Av. André Tosello 209, Campinas, SP 13083-886, Brazil
[b] Institute of Computing, University of Campinas, Av. Albert Einstein 1251, Campinas, SP 13083-852, Brazil

ARTICLE INFO

Keywords:
Fruit detection
Yield prediction
Computer vision
Deep learning

ABSTRACT

Agricultural applications such as yield prediction, precision agriculture and automated harvesting need systems able to infer the crop state from low-cost sensing devices. Proximal sensing using affordable cameras combined with computer vision has seen a promising alternative, strengthened after the advent of convolutional neural networks (CNNs) as an alternative for challenging pattern recognition problems in natural images. Considering fruit growing monitoring and automation, a fundamental problem is the detection, segmentation and counting of individual fruits in orchards. Here we show that for wine grapes, a crop presenting large variability in shape, color, size and compactness, grape clusters can be successfully detected, segmented and tracked using state-of-the-art CNNs. In a test set containing 408 grape clusters from images taken on a trellis-system based vineyard, we have reached an F_1-score up to 0.91 for instance segmentation, a fine separation of each cluster from other structures in the image that allows a more accurate assessment of fruit size and shape. We have also shown as clusters can be identified and tracked along video sequences recording orchard rows. We also present a public dataset containing grape clusters properly annotated in 300 images and a novel annotation methodology for segmentation of complex objects in natural images. The presented pipeline for annotation, training, evaluation and tracking of agricultural patterns in images can be replicated for different crops and production systems. It can be employed in the development of sensing components for several agricultural and environmental applications.

1. Introduction

Automation in agriculture is particularly hard when compared to industrial automation due to field conditions and the uncertainty regarding plant structure and outdoor environment. That creates a need for systems able to monitor structures as plants and fruits in a fine-grained level (Kirkpatrick, 2019). Proper detection and localization for such structures are critical components for monitoring, robotics and autonomous systems for agriculture (Duckett et al., 2018).

Accurate fruit detection and localization are essential for several applications. Fruit counting and yield estimation are the more immediate ones. Precision agriculture applications, accounting for management of inter and intra-field variability, can be derived if detection data is properly localized in space. Fruit detection can also be a preliminary step for disease and nutrient deficiency monitoring (Barbedo, 2019) and a crucial component on actuation, for example, automated spraying and harvesting may be an important application considering the declining in agricultural labor force (Roser, 2019). Beyond farms, fruit detection can be employed in field phenotyping, aiding plant research and breeding programs (Kicherer et al., 2017; Rose et al., 2016).

Off-the-shelf RGB cameras and computer vision can provide affordable and versatile solutions for fruit detection. State-of-the-art computer vision systems based on deep convolutional neural networks (LeCun et al., 2015) can deal with variations in pose, shape, illumination and large inter-class variability (He et al., 2016; Krizhevsky et al., 2012; Simonyan and Zisserman, 2015), essential features needed for robust recognition of complex objects in outdoor environments. Recent researches (Bargoti and Underwood, 2017a; Sa et al., 2016) have shown that the Faster R-CNN (region-based convolutional neural network) architecture (Ren et al., 2015) can produce accurate results for a large set of fruits, including peppers, melons, oranges, apples, mangoes, avocados, strawberries and almonds. Detection results can be integrated by data association approaches, by employing object tracking or mapping, to perform fruit counting for rows in the crop field (Liu et al., 2019).

These previous detection systems identify individual objects by

* Corresponding author.
E-mail addresses: thiago.santos@embrapa.br (T.T. Santos), sandra@ic.unicamp.br (S. Avila).

https://doi.org/10.1016/j.compag.2020.105247
Received 13 August 2019; Received in revised form 23 January 2020; Accepted 26 January 2020
Available online 07 February 2020

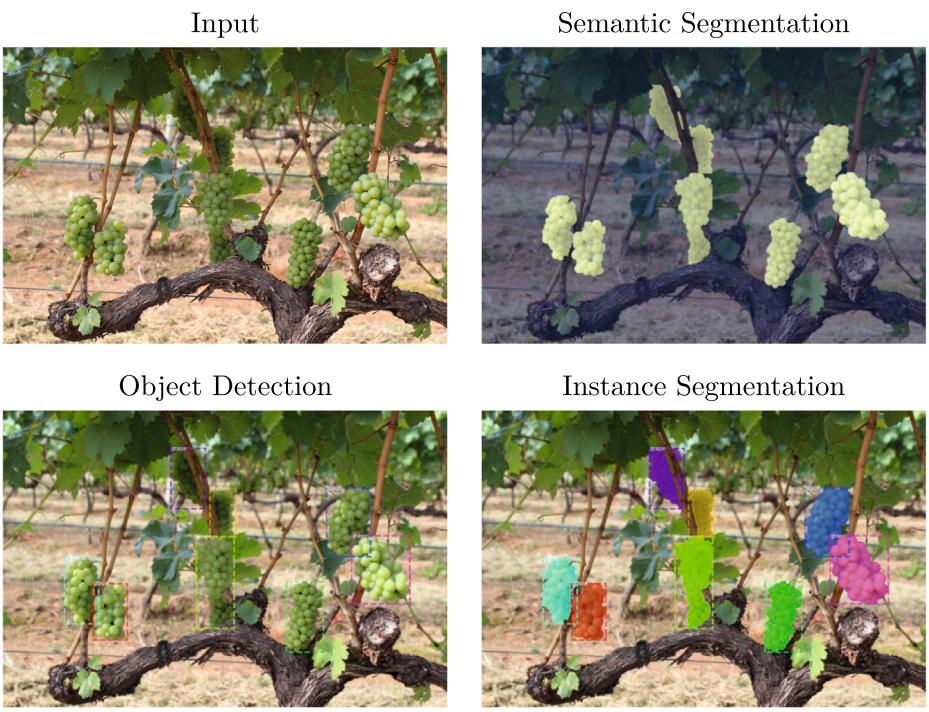

Fig. 1. Dataset entry example. Visual recognition can be stated as three different problems: (i) semantic segmentation (a pixel classification problem for fruit/non-fruit), (ii) object detection (fruit localization by bounding boxes) and (iii) instance segmentation. The most challenging variation, instance segmentation, is object detection and pixel attribution combined (each pixel is attributed to one of the detected objects or to the background) (Lin et al., 2014).

rectangular bounding boxes, as seen in Fig. 1. Such boxes, if well fitted to the fruit boundaries, could provide estimations of fruit shape and space occupancy for fruits presenting a regular shape as oranges and apples (circular shape). However, for grape clusters, rectangular boxes would not properly adjust to the berries. A step further beyond object detection is *instance segmentation* (Lin et al., 2014): the fruit/non-fruit pixel classification combined with instance assignment (Fig. 1). Instance segmentation can properly identify berries pixels in the detection box, providing finer fruit characterization. Also, *occlusions* by leaves, branches, trunks and even other clusters can be properly addressed by instance segmentation, aiding on robotic manipulation and other automation tasks.

To approach fruit instance segmentation as a supervised machine learning problem, we need datasets that capture the variations observed in the field. Wine grapes present large variations in shape, size, color, and structure, even for the same grape variety, contrasting to citrus and apples. Also, the dataset has to provide *masks* for the individual clusters, isolating grapes from background pixels and from occluding objects. We also need a neural network architecture able to perform object detection and pixel classification simultaneously. Thus, the present work introduces the following contributions:

1. a new methodology for image annotation that employs interactive image segmentation (Noma et al., 2012) to generate object masks, identifying background and occluding foreground pixels;
2. a new public dataset[1] for grape detection and instance segmentation, comprising images, bounding boxes and masks – this dataset is composed by images of five different grape varieties taken on field (Fig. 1);
3. an evaluation of two deep learning detection architectures for grape detection: Mask R-CNN (He et al., 2017), a convolutional framework for instance segmentation that is simple to train and generalizes well (Liu et al., 2018), and YOLO (Redmon et al., 2016), a single-stage network that can detect objects without a previous region-proposal stage (Huang et al., 2017) – such evaluation allows a comparison between instance segmentation and box-based object detection approaches;
4. a fruit counting methodology that employs three-dimensional association to integrate and localize the detection results in space, avoiding multiple counting, addressing occlusions and accumulating evidence from different images to confirm detections.

[1] Available at doi:https://doi.org/10.5281/zenodo.3361736 (Santos et al., 2019).

2. Related work

As seen in computer vision applications on other fields, classic machine learning and pattern recognition have been replaced by modern deep learning techniques, which can address the enormous variability in object appearance, as shortly described in the following sections.

2.1. Earlier works: feature engineering and machine learning

Earlier works in fruit detection employed the classic *feature engineering* approach: human-designed descriptors based on color, geometric and texture features. Using such features, machine learning techniques such as Bayesian classifiers, support vector machines and clustering were applied to perform fruit detection and classification. Gongal et al. (2015) presented an extensive review of the works employing this approach. Dunn and Martin (2004) presented one of the earliest works to employ image processing for grape detection. They used color thresholding to detect “grape pixels” in images showing mature *Cabernet Sauvignon* clusters in vines. A white screen was placed behind the canopy to create a regular background.

Nuske et al. (2011) presented a computer vision methodology intended for realistic field operation without background control. Their multi-stage method employed Loy and Zelinsky’s Radial Symmetry Transform (Loy and Zelinsky, 2003) to find berry candidates, further filtered by a K-nearest neighbors classifier using color and texture features. In the last step, neighboring berries were grouped in clusters, eliminating isolated berries (likely false-positives). For a set of 2,973 berries, their system reached 63.7% for recall and 98.0% for precision overall (the set included berries of three grape varieties). The authors performed linear regression for the berries count found for individual vines, finding a 0.74 correlation score for crop weight.

In a further work, Nuske et al. (2014) added a data association component based on visual odometry (Scaramuzza and Fraundorfer, 2011) to avoid double-counting and to estimate the spatial distribution of yield. They proposed a new berry detector for a particular flash-based setting for night imaging developed by them and evaluated other image features for berry classification: SIFT (Lowe, 2004) and FREAK (Alahi et al., 2012). Nuske et al. (2014) stated that segmentation of berries clusters (grape clusters) is challenging because of occlusion and touching clusters; after some experiments with 3-D modeling, the authors chose to perform yield estimation using berry counting. They performed controlled imaging and reported variations in results possibly caused by illumination and imaging differences.

2.2. Deep learning-based works

Earlier works present issues that foreshadow the advantages and power of convolutional neural networks (CNNs). These networks learn effective representations for a given machine learning task, replacing feature engineering (Bengio et al., 2013). Systematically, deep learning approaches are being adopted in fields presenting image-based perceptual problems, and agricultural applications are no exception (Kamilaris and Prenafeta-Boldú, 2018).

CNN's *invariance to local translation* give vision systems robustness in situations where a feature's presence is more important than its exact location (Goodfellow et al., 2016). As an example, Nuske et al. (2014) reported that variations in the berry candidate location by detection affected their berry classification. CNNs are also able to encode variance regarding pose, color and illumination, if the training data presents sufficient examples of such variation, which relieves the need for controlled imaging, illumination and camera settings. The first attempts employed CNNs to perform pixel classification, followed by additional steps to segment individual fruits (Bargoti and Underwood, 2017b; Chen et al., 2017). Further, these earlier approaches were replaced by *end-to-end object detection* (Bargoti and Underwood, 2017a; Liu et al., 2019; Sa et al., 2016) based on the popular Faster R-CNN architecture (Ren et al., 2015).

Sa et al. (2016) employed transfer learning, using a VGG16 network (Simonyan and Zisserman, 2015) pre-trained using ImageNet (Deng et al., 2009) (VGG16 is the *perceptual backbone* in the Faster R-CNN architecture). They reached F_1-scores up to 0.83 in tests on sweet pepper and rock melon, using a dataset of images captured in a greenhouse, and presented similar performance for smaller datasets of strawberry, apple, avocado, mango and orange images retrieved from Google Images Search. The authors also fused RGB and Near Infrared (NIR) data in four-channel arrays, showing that the CNN paradigm can easily benefit from multi-spectral imaging.

Bargoti and Underwood (2017a) also employed the Faster R-CNN architecture for fruit detection. They produced datasets from images captured in orchards by a robotic ground vehicle for apples and mangoes, and a dataset for almonds, also in orchards, but using a hand-held DSLR camera (digital single-lens reflex). Employing image augmentation strategies on training, the authors reached F_1-scores up to 0.90 for mangoes and apples and 0.77 for almonds. A surprising result reported by the authors is that transfer learning between farms (same crop) or between orchards of different crops showed little advantage compared to ImageNet transfer learning. Bargoti and Underwood state such result increase the body of evidence showing ImageNet features applicability for a broad range of tasks.

In Faster R-CNN, detection is performed in two stages. The first stage uses a *region proposal network*, an attention mechanism developed as an alternative to the earlier sliding window based approaches. In the second stage, bounding box regression and object classification are performed. Faster R-CNN is fairly recognized as a successful architecture for object detection, but it is not the only *meta-architecture* (Huang et al., 2017) able to reach state-of-the-art results. Another group of architectures is the *single shot detector* (SSD) meta-architecture (Huang et al., 2017; Liu et al., 2016), single feed-forward convolutional networks able to predict classes and bounding boxes in a single stage. The YOLO (*You Only Look Once*) networks, proposed by Redmon et al. (2016) and Redmon and Farhadi (2017), are examples of the SSD family.

Grape clusters present larger variability on size, shape and compactness compared to other fruits like peppers, apples or mangoes (Bargoti and Underwood, 2017a; Sa et al., 2016). A focus on berry detection, such as in Nuske et al. (2011, 2014), can be seen as a way to circumvent grape cluster variability, performing yield prediction over berry counting, consequently bypassing the grape cluster segmentation problem. CNNs can learn representations of complex visual patterns (Goodfellow et al., 2016), so are an interesting alternative for grape cluster detection. However, object detection using bounding boxes could be insufficient for yield prediction applications, considering the enormous variability in grape clusters' shapes and compactness. On the other hand, semantic segmentation (the classification of pixels as fruit or background) could also be inadequate, considering the severe occlusion between fruits observed in orchards (Bargoti and Underwood, 2017a). *Instance segmentation* (Fig. 1), the combined task of object detection (where are the grape clusters?) and pixel classification (this pixel belongs to which cluster?), is an alternative machine learning task formulation for yield prediction and automated harvesting applications.

Mask R-CNN (He et al., 2017) is a derivation of Faster R-CNN able to perform instance segmentation, jointly optimizing region proposal, bounding box regression and semantic pixel segmentation. However, differently of object detection in which rectangular bounding boxes annotations are sufficient for training, instance segmentation needs image pixels to be properly attributed to an instance or to the background in the training dataset for supervised learning. In Section 3, we describe a methodology for fruit instance segmentation based on Mask R-CNN, including a novel instance annotation tool for objects of complex shape. We compare YOLO and Mask R-CNN results on wine grape cluster detection, and we evaluate Mask R-CNN results on cluster instance segmentation.

Fruit detection in single images can be the *perceptual step* in a fruit counting system, but without some sort of integration of the information produced for the orchard, accurate prediction of yield is not possible. Liu et al. (2019), extending the work in Bargoti and Underwood (2017a), integrated the fruit detection results in image sequences (video frames) performing *object tracking*. Employing the bounding box centers as observations, the authors implemented an object tracker based on the Kanade-Lucas-Tomasi algorithm (optical flow), Kalman filters and the Hungarian Assignment algorithm, tracking fruits in video frame sequences. To address issues caused by missing detections and occlusions, they performed *structure-from-motion*, recovering three-dimensional information using the box centers and their inter-frame correspondence. Associating fruit locations in 3-D and the CNN detection in 2-D frames, Liu et al. (2019) integrated data from a camera moving along a mango orchard row, avoiding double-counting from the same fruit observed in different frames, addressing occlusions and localizing yield information in space. Similarly, we propose a simple but effective *spatial registration step* for fruit tracking and counting, also employing 3-D association from structure-from-motion data.

3. Materials and methods

The proposed methodology introduces a new public dataset for image-based grape detection, including a novel method for interactive mask annotation for instance segmentation (Section 3.1). Three neural networks are trained and evaluated for fruit detection: Mask R-CNN (He et al., 2017), YOLOv2 (Redmon et al., 2016) and YOLOv3 (Redmon and Farhadi, 2018) (Section 3.2). Evaluation measures for semantic segmentation, object detection, and instance segmentation variants are presented in Section 3.4. Section 3.5 presents our approach for spatial integration.

Table 1
General information about the dataset: the grape varieties and the associated identifying prefix, the date of image capture on field, number of images (instances) and the identified grapes clusters.

Prefix	Variety	Date	Images	Boxed clusters	Masked clusters
CDY	*Chardonnay*	2018–04-27	65	840	308
CFR	*Cabernet Franc*	2018–04-27	65	1,069	513
CSV	*Cabernet Sauvignon*	2018–04-27	57	643	306
SVB	*Sauvignon Blanc*	2018–04-27	65	1,317	608
SYH	*Syrah*	2017–04-27	48	563	285
Total			300	4,432	2,020

3.1. The dataset

The *Embrapa Wine Grape Instance Segmentation Dataset* (WGISD) is composed by 300 RGB images showing 4,432 grape clusters from five different grape varieties, as summarized in Table 1. All images were captured from a single winery, that employs *dual pruning*: one for shaping (after previous year harvest) and one for production, resulting in canopies of lower density. No pruning, defoliation or any intervention in the plants was performed specifically for the dataset construction: the images capture a real, trellis system-based wine grape production. The camera captures the images in a frontal pose, that means the camera principal axis is approximately perpendicular to the wires of the trellis system and the plants rows. As seen in Table 1, the *Syrah* images were taken in a different field visit, one year earlier, and consist a smaller set if compared to the other four varieties. A presents a detailed description for the dataset, following the guidelines proposed by Gebru et al. (2018) for dataset characterization, and including information about cameras, field location, pre-processing and file formats. The WGISD is publicly available (Santos et al., 2019) under the CC BY-NC 4.0 (Attribution-NonCommercial 4.0 International) license.

To be employed on supervised instance segmentation training, WGISD has to provide a set of *masks* that properly segment grape clusters. WGISD provides binary masks for 2,020 clusters from the 4,432 total, as seen in Table 1. Mask annotation for instance segmentation is a laborious task that requires custom tools to allow the annotation of hundreds of images in limited time. The VGG Image Annotator (VIA) (Dutta et al., 2016) is a popular tool used by the computer vision community. It allows users to mark objects of interest using rectangles, circles, ellipses or polygons. In an interesting attempt to automatize annotation, Acuna et al. (2018) proposed an interactive tool that uses a neural network (Polygon-RNN++) to predict the next vertex in polygonal annotations.

In WGISD construction, the accurate annotation of complex objects in natural scenes using polygonal shapes proved to be extremely laborious, even when employing the vertex prediction facilities from Polygon-RNN++. To relieve the annotation process, we have created an annotation tool based on interactive image segmentation by graph matching, as proposed by Noma et al. (2012). This method starts from an over-segmentation, produced by the watershed algorithm (Vincent and Soille, 1991), to create an *attributed relational graph* (ARG) representation for the image – G_i. Then, the user can freely mark the image using *scribbles*. Such marks are used to create a *model graph* G_m, a labeled ARG. Exploiting the spatial relations among ARGs vertices, a match is computed between the model graph G_m and the input image graph G_i, allowing the propagation of labels from G_m to G_i.

Fig. 2 shows an example of grape annotation. The dataset was previously annotated for object detection using standard rectangular bounding boxes (see Appendix A for details). The instance annotation tool uses the bounding boxes as inputs, displaying each grape cluster for an interactive image segmentation procedure by graph matching (Fig. 2(a)). An annotator can draw scribbles, freely marking pixels that should be considered part of the grape cluster and pixels that are part of the background or occluding foreground objects (Fig. 2(b)). The graph matching-based algorithm uses the scribbles to produce a segmentation, propagating the labels from the model to the input image (Fig. 2(c)). The tool allows the scribble marking and graph matching steps to be repeated by the user until a reasonable annotation is achieved. Finally, the grape pixels are stored as masks for supervised instance segmentation learning. Readers interested in a detailed description of the graph matching algorithm should refer to Noma et al. (2012).

3.2. The perceptual step: CNN architectures

Mask R-CNN (He et al., 2017) is a consolidation of a long sequence of works developed by He, Dollár, Girshick and colleagues. This network is essentially the combination of a Faster R-CNN object detector (Ren et al., 2015) and a *fully convolutional network* (FCN) (Shelhamer et al., 2017) for semantic segmentation, providing a complete, end-to-end, instance segmentation solution. The Faster R-CNN is also a combination of two architectures: a *region proposal network* (RPN) and an object detector, the Fast R-CNN (Girshick, 2015). RPN works as an attention mechanism, finding *anchors* in the feature space, rectangular boxes that can contain objects of interest (Fig. 3). The Fast R-CNN is composed of a softmax object classifier and a per-class bounding box regressor (Fig. 4). The Mask R-CNN employs as feature extractor a *feature pyramid network* (FPN) (Lin et al., 2017), an architecture able to

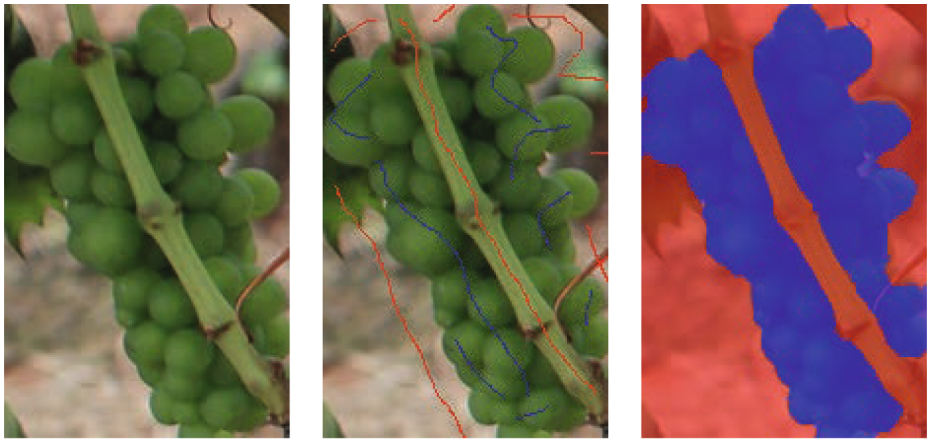

Fig. 2. Instance annotation using interactive image segmentation by attributed relational graphs. (a) Grape cluster delimited using a standard bounding box annotation. (b) Scribbles drawn by the user (blue for grapes, red for background or foreground structures). (c) Segmentation produced by the graph matching procedure. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

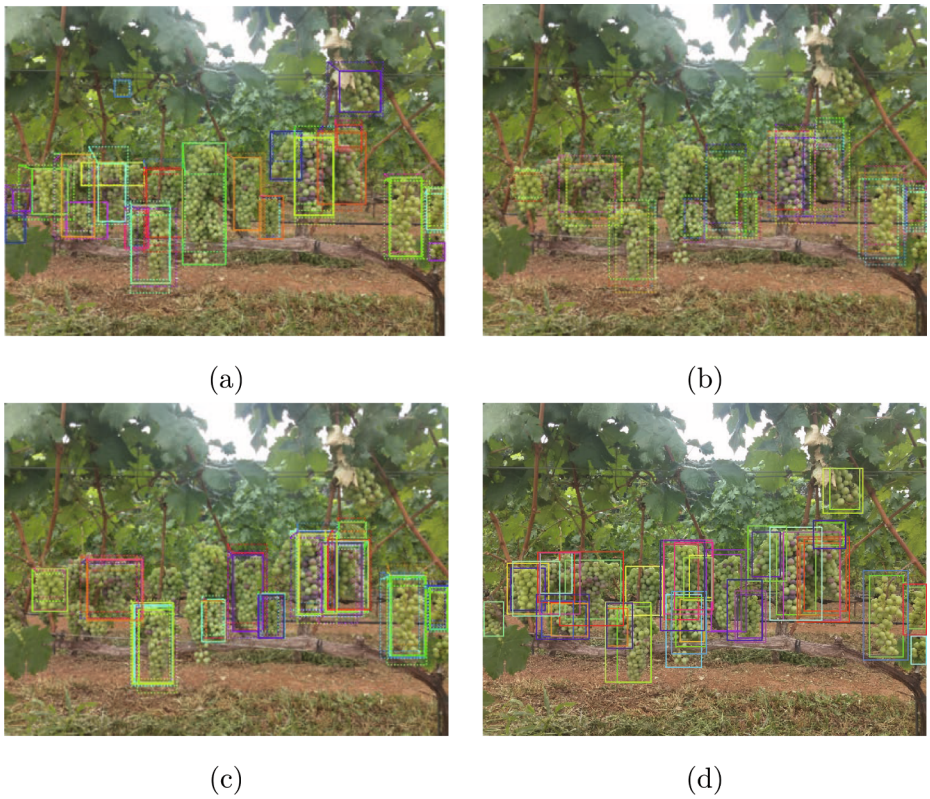

Fig. 3. RPN network under action. (a) Targets for training the RPN, built from the training set - note the anchors (dashed lines) and the location and size deltas (solid lines). (b) Subset of the top rated anchors (few anchors shown to improve visualization for the reader). (c) Subset of the top anchors, after refinement. (d) Final regions found by the RPN for the image after non-max suppression.

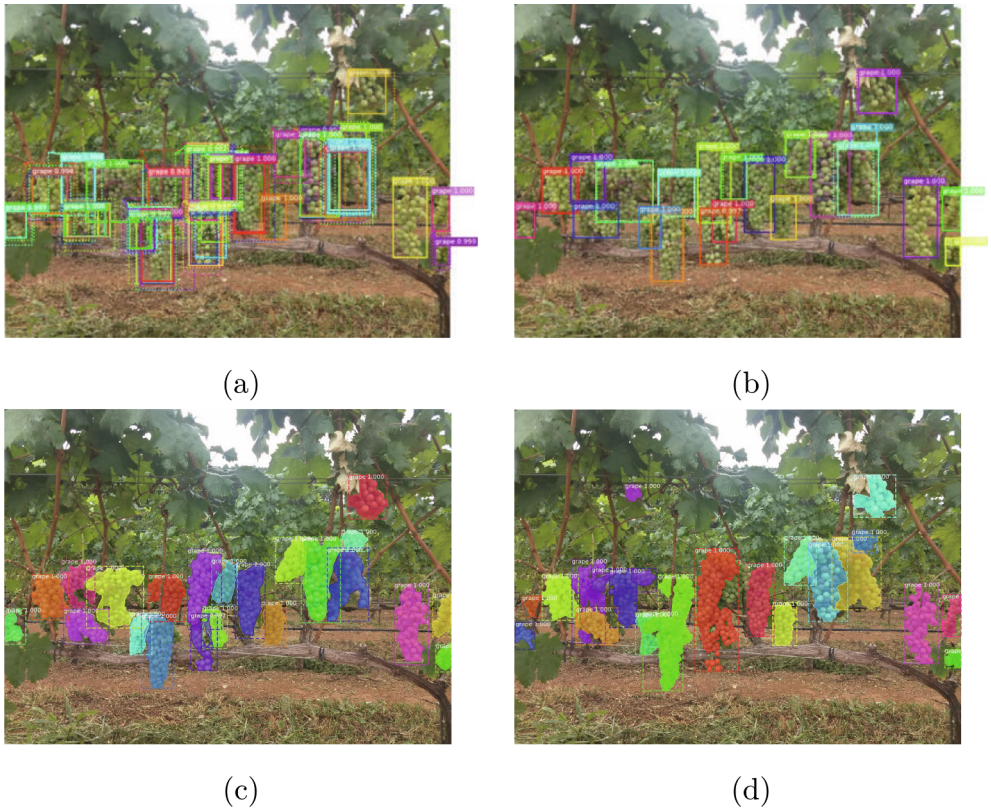

Fig. 4. Mask R-CNN under action. (a) Class-specific bounding box refinement. (b) Results after low-confidence filtering and non-max suppression. (c) Final results after FCN pixel classification (d) Ground-truth.

Table 2
Training and test sets sizes, stratified by grape variety. Images were randomly selected for each set in an 80–20% split (approximately). In the *Masked clusters* column, the number in parentheses corresponds to the number of *masked* images available.

	Variety	Images	Boxed clusters	Masked clusters
Train./Val.	CDY	50	660	227 (18)
	CFR	55	910	418 (27)
	CSV	48	532	241 (23)
	SVB	51	1034	502 (24)
	SYH	38	446	224 (18)
	Total	242	3582	1612 (110)
Test	CDY	15	180	81 (6)
	CFR	10	159	95 (6)
	CSV	9	111	65 (5)
	SVB	14	283	106 (5)
	SYH	10	117	61 (5)
	Total	58	850	408 (27)

create semantic feature maps for objects at multiple scales, built over a ResNet (He et al., 2016).

Another approach to object detection is to predict the locations and the objects' class in a single step, in order to avoid a previous region proposal procedure. Huang et al. (2017) refer to this approach as *single shot detector meta-architecture*, and the YOLO networks proposed by Redmon et al. (2016, 2017) are prominent members of this family. In the YOLO networks, the image is split into a fixed grid of $S \times S$ *cells*. A cell is responsible for performing a detection if an object center is over it. Each cell is associated to B boxes, composed by 5 values representing the object center (c_x, c_y), the object width and height and a *confidence score* that represents the model confidence that the box contains an object and also the accuracy of the box boundaries regarding the object. The box also includes C conditional class probabilities, one to each class of objects. Consider, for example, a 7×7 grid of cells ($S = 7$), where each cell predicts $B = 2$ boxes for 20 different classes of object ($C = 20$). The YOLO network will produce a $7 \times 7 \times 30$ output tensor. This means a $B{\cdot}5 + C$ vector for each one of the 49 cells. The training step tries to minimize a loss function defined over such a tensor, performing detection and classification in a single step. The YOLOv2 and YOLOv3 networks have a few differences, mainly regarding their feature extraction convolution part. YOLOv3 presents a deeper convolutional network that incorporate some state-of-the-art techniques such as residual networks (He et al., 2016), skip connections and multi-scaling (similar to FPNs). YOLOv3 classification is based in multi-label classification instead of the softmax employed by YOLOv2, allowing the former able to deal with multi-class problems.

3.3. Training

Table 2 shows the splitting between training/validation and test sets. The images were assigned randomly to each set in an 80–20% proportion. It is important to note that *Cabertnet Franc* and *Sauvignon Blanc* varieties presented a higher number of clusters per image. In Section 4, we will show that although the differences in the numbers of images and clusters, the results are very similar for all five grape varieties. For instance segmentation, a set of 110 images presenting masks is available for training. We have split it into an 88 images training set (1,307 clusters) and a validation set composed of 22 images (305 clusters).

We employed image augmentation to mitigate overfitting (Chollet, 2017), adopting transformations of the original images that could simulate field conditions: differences in lighting, camera focus, noise or dirty lenses. We applied Gaussian blur, contrast normalization, additive Gaussian noise and pixel dropouts[2] using the `imgaug` library (Jung, 2019). These augmentations were randomly selected and ordered in such a way that *different transformations* were applied for each source image. We have applied 20 random augmentations for each image, as shown in Fig. 5, which produced a rich set of variations, potentially reflecting real field conditions.

We employed the Keras/TensorFlow-based implementation for Mask R-CNN developed by Matterport, Inc., publicly available at GitHub (Matterport, Inc., 2018). The network was initialized using the weights previously computed for the COCO Dataset (Lin et al., 2014). No layer was frozen during training, so all weights could be updated by the training on the grapes dataset. Due to GPU memory limitations, to allow multiple images per batch, input images are resized to $1024 \times 1024 \times 3$ tensors, preserving aspect ratio by applying zero padding as needed. Two feature extraction architectures were evaluated: ResNet-101 and the shallower ResNet-50 (He et al., 2016). For the YOLO networks, we employed Darknet, the original implementation developed by Redmon et al. (2016), initialized using pre-trained weights from ImageNet (Deng et al., 2009). In our single-class grape detection case, $C = 1$. Training was performed on a computer containing a single NVIDIA TITAN Xp GPU (12 GB memory), an Intel i7-x990 CPU and 48 GB RAM, and running Ubuntu 18.04 LTS. For Mask R-CNN, training was performed in approximately 10 h (100 epochs, around 6 min per epoch). YOLO training (v2 and v3) spent four days using the same hardware.

3.4. Evaluation

The WGISD dataset allows evaluations for the semantic segmentation, object detection and instance segmentation problems. This work will present results using the standard metrics of *precision* (P), *recall* (R), and their harmonic mean (F_1), as usual in the information retrieval literature:

$$P = \frac{N_{\text{tp}}}{N_{\text{tp}} + N_{\text{fp}}}, \tag{1}$$

$$R = \frac{N_{\text{tp}}}{N_{\text{tp}} + N_{\text{fn}}}, \text{ and} \tag{2}$$

$$F_1 = \frac{2{\cdot}P{\cdot}R}{P + R}. \tag{3}$$

These measurements depend on the number of *true positives* (N_{tp}), *false negatives* (N_{fn}) and *false positives* (N_{fp}), which need to be properly defined for each type of problem:

- For *semantic segmentation*, we are considering just one class (grape) and pixel classification. In this case, we employ the masked images in the test set for evaluation, where the grape pixels are properly marked (27 images, 408 grape clusters). $N_{\text{tp}}^{\text{seman}}$ is the number of pixels correctly classified as grape pixels according to the ground truth, $N_{\text{fn}}^{\text{seman}}$ the number of grape pixels incorrectly classified as non-grape pixels, and $N_{\text{fp}}^{\text{seman}}$ the number of non-grape pixels wrongly reported as grape ones by the classifier. Such three measures allow the computation of P_{seman} and R_{seman}, respectively precision and recall, for the semantic segmentation problem.
- In *object detection*, each grape cluster instance is localized by a rectangular bounding box. A hit or a miss is defined by a one-to-one correspondence to ground truth instances, obeying an *intersection over union* (IoU) threshold computed using the rectangular areas and their intersections. $N_{\text{tp}}^{\text{box}}$ is the number of correctly predicted instances (bounding boxes) and $N_{\text{fn}}^{\text{box}}$ and $N_{\text{fp}}^{\text{box}}$ are similarly defined.

[2] Similar to "pepper" noise – see `imgaug` documentation for details (Jung, 2019).

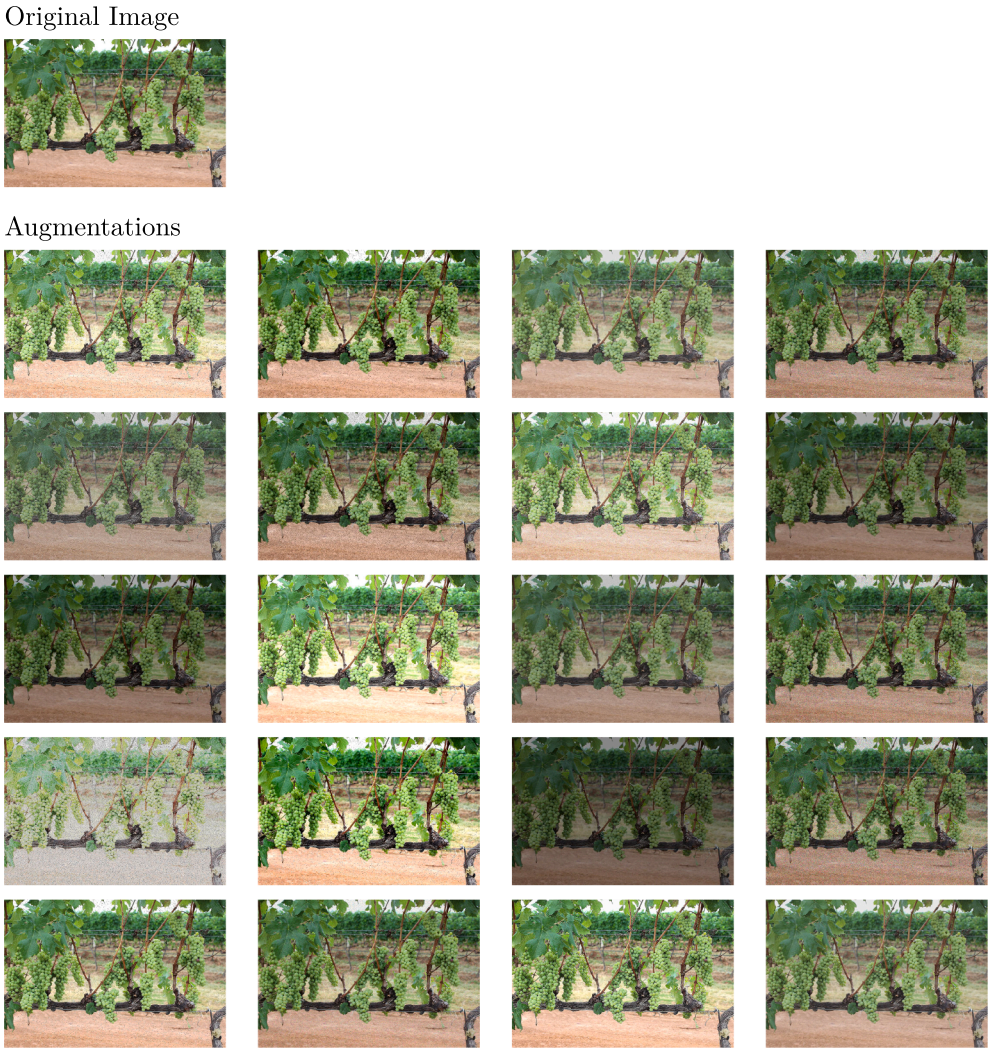

Fig. 5. Image augmentations produced for a *Cabernet Franc* image. The randomized pipeline of transformations produces variations in lighting, contrast and noise. Dirty lenses are emulated using pixel dropouts..

These three measures give the values for P_{box} and R_{box}, respectively precision and recall, for the object detection problem and the evaluation is performed for the entire test set (58 images, 837 grape clusters).

- *Instance segmentation* follows the same instance-based logic as in object detection, but IoU is computed using the areas and intersections of the *masks* instead of rectangular bounding boxes. Again, we are limited to the masked images in the test set: 27 images containing 408 clusters. The measures are P_{inst} and R_{inst} for instance segmentation precision and recall, respectively.

Mask R-CNN results can be evaluated for the three problems, but the YOLO-based results are just evaluated regarding object detection.

3.5. Spatial registration: 3-D association

Structure-from-Motion (SfM) (Hartley and Zisserman, 2003) is a fundamental achievement in computer vision and a core component in modern photogrammetry. It solves the camera pose and scene geometry estimation simultaneously, employing only image matching and bundle adjustment (Triggs et al., 2000), and finding three-dimensional structure by the motion of a single camera around the scene (or from a set of independent cameras). In a previous work (Santos et al., 2017), we showed that SfM can recover vine structure from image sequences on vineyards. It is an interesting alternative to integrate image data from different camera poses registering the same structures in space. Similarly to Liu et al. (Liu et al., 2019), we use 3-D data from the COLMAP SfM software (Schönberger and Frahm, 2016) to perform spatial registration, integrating the fruit instance segmentation data produced by the perceptual CNN-based step.

Consider the directed graph $G = (V, E)$, where V is a set of nodes $u_{i,j}$ representing the j-th instance found by the neural network in the i-th frame. Consider the set of $\mathcal{X} = \{\mathbf{X}_k\}_{k=1..M}$ of M three-dimensional points $\mathbf{X}_k$ found by the SfM procedure. We create an oriented edge $(u_{i,j}, v_{i',j'}) \in E$, considering $i < i'$, if there is a 3-D point $\mathbf{X}_k$ that projects to the instance j in frame i and to instance j' in frame i'. In other words, there is a link between instances from two different frames if there is a three-dimensional point whose 2-D projections are contained in the masks associated to these instances, evidence they could be observing the same object in the 3-D world.

Fig. 6 (a) shows the graph G. Each node represents an instance (a grape cluster) and a column of nodes represents all instances found in a frame i. Neighboring columns represent successive frames i and i', $i < i'$. G is a directed graph, with oriented edges indicating instance association between frames. Each edge has a *weight* $w[u_{i,j}, v_{i',j'}]$ that indicates the total number of three-dimensional points that links the two instances, accumulating the evidence that associates instance j in frame i to the instance j' in i'.

The structure of G is affected by occlusions: if changes in camera

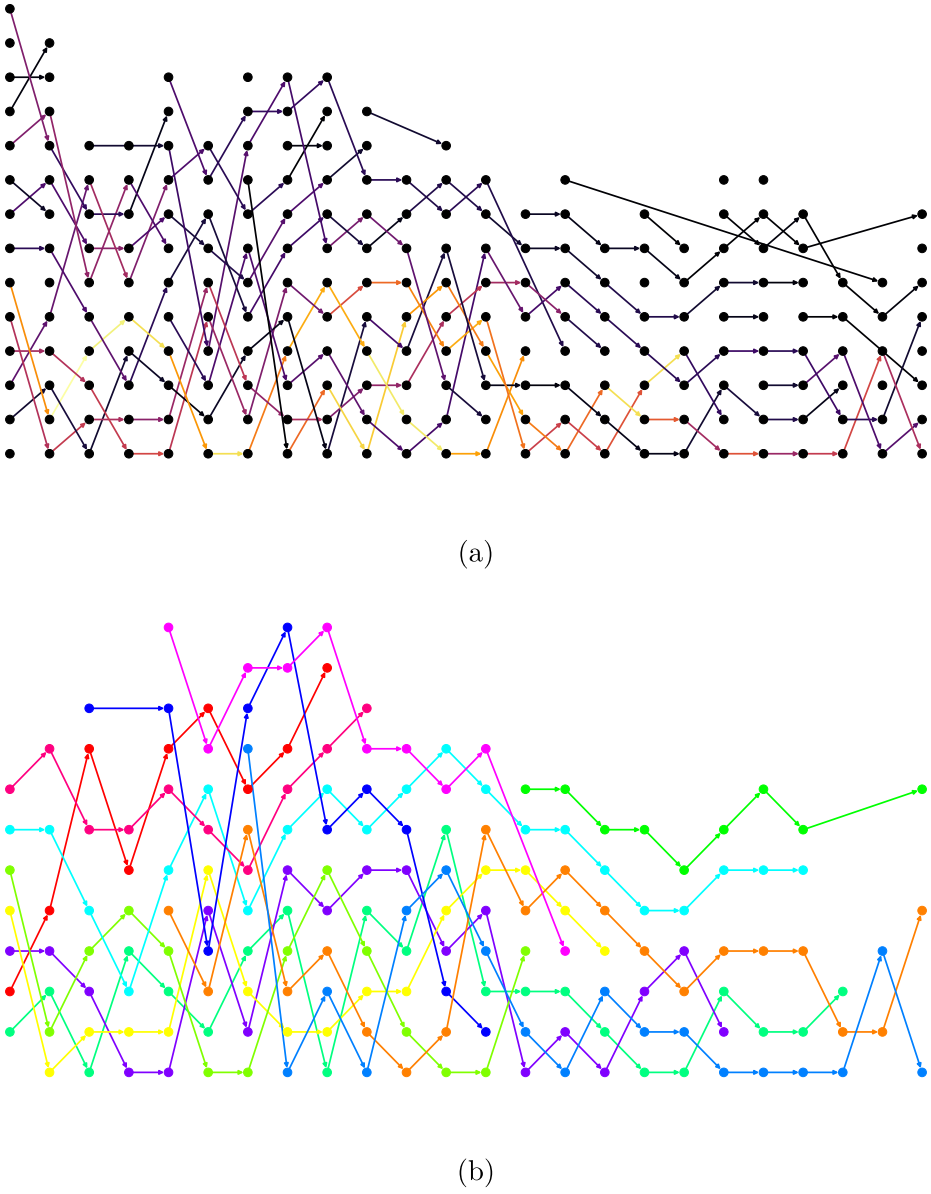

Fig. 6. Instance matching and tracking using 3-D assignment. Each column represents the instances found by a neural network in a video frame. (a) Matching found by 3-D assignment – the edges are strong (warm colors) as the number of 3-D points linking the instances. (b) Instance tracking found by looking for the deepest paths in the graph – each color represents an individual grape cluster. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Instance segmentation results for Mask R-CNN. This evaluation was performed in the masked test set, considering a confidence level of 0.9 for the *grape* class.

IoU	AP	P_{inst}	R_{inst}	F_1
0.3	0.855	0.938	0.892	0.915
0.4	0.822	0.923	0.877	0.899
0.5	0.743	0.869	0.826	0.847
0.6	0.635	0.799	0.760	0.779
0.7	0.478	0.696	0.662	0.678
0.8	0.237	0.485	0.461	0.472
0.9	0.008	0.070	0.066	0.068

pose make two or more clusters to occlude each other, then two or more edges will be incident to a node $u_{i,j}$. Similarly, when the camera movement reveals two or more occluded clusters, two or more edges will flow from the same node. We filter the edges in E in such a way that, for each node, there is up to one incident edge and up to one departing edge. The filtering strategy is simple: the *heaviest* (maximum weight w) edge is kept. The intuition behind this strategy is it would favor the *occluding* grape cluster while the *occluded* one is tracked by an edge spanning many frames – that means $(u_{i,j}, v_{i',j'})$ where $i' > i + 1$. These edges spanning many frames also help with the relocalization of grapes occluded by other objects in the scene (leaves, trunks, etc.) missed by the neural network in some frames. After edge filtering, nodes are sorted by their frame index i and, for each node $u_{i,j}$, we find the longest path in G using depth-first search on edges, corresponding to the track of one grape cluster along the frame sequence. Too short paths (we use a threshold of 5 edges) are filtered out, an attempt to remove false positives from the perceptual stage by integrating evidence from multiple frames. Fig. 6 (b) illustrates the final set of longest and disjoints paths, where different colors discriminate different tracks (different grape clusters). The number of paths is an estimation of the total number of grape clusters in the entire image sequence.

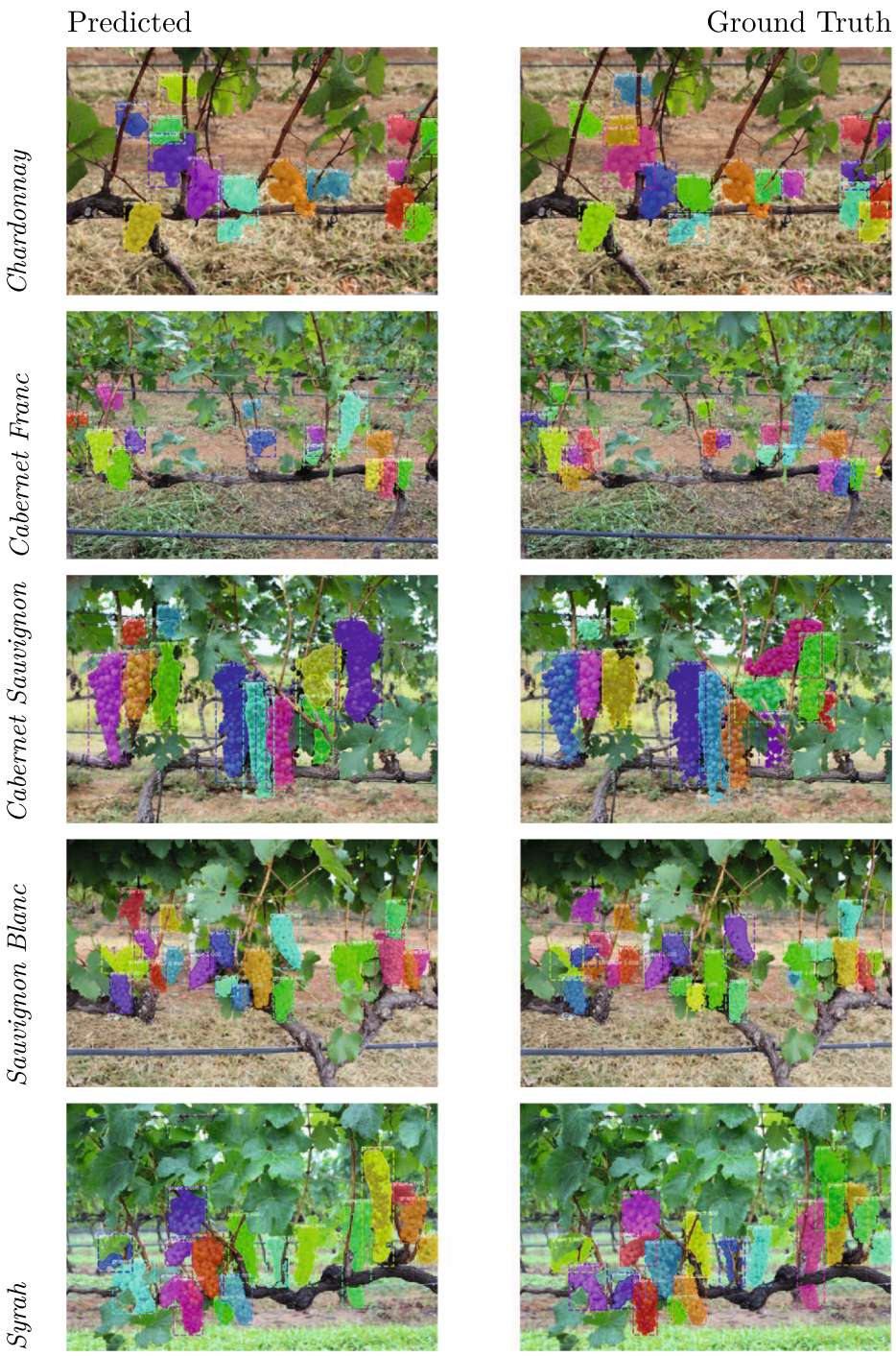

Fig. 7. Some instance segmentation results produced by Mask R-CNN, one example for each grape variety. (Left) Predictions by the network. (Right) Ground truth. Same color does not mean assignment between prediction and ground truth. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4. Results

The validation set was employed to select the best models for further evaluation on the test set. For the Mask R-CNN, the ResNet 101 feature extraction backbone produced the best results. Table 3 presents the evaluation of predictions produced by Mask R-CNN for instance segmentation, considering the masked test set (408 clusters in the ground truth) and confidence threshold of 0.9 for the grape class.[3] The table shows the precision and recall measures for seven different values of IoU, from 30% to 90%. The corresponding values for F_1 score and *average precision*[4] (AP) as defined in Pascal VOC Challenge (Everingham

[3] In the experiments, Mask R-CNN did not exhibit great sensibility to the (*footnote continued*) confidence threshold. We have tested 0.5, 0.7, 0.9 and 0.95, all presenting very similar results, and F_1 variations inferior to 0.005.

[4] The AP summarizes the shape of the precision/recall curve, and it is defined as the mean precision at a set of equally spaced recall levels. See Everingham et al. (2010) for details.

Fig. 8. Divergence between predicted segmentation and the ground truth. (Left) Predictions by the network – red clusters are false positives, green clusters are true positives. (Right) Ground truth – blue clusters are false negatives. Disagreement in segmentation creates false negatives and false positives, despite correct detection of grape berries. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4
Semantic segmentation by Mask R-CNN. The first lines show evaluation for semantic segmentation (grape/background) for each image in the test set, stratified by variety for comparison. The last line shows the evaluation for the entire test set (computed by accumulation of true positives, false positives and false negatives values).

Image	P_{seman}	R_{seman}	F_1
CDY 2043	0.959	0.902	0.929
CDY 2051	0.961	0.871	0.913
CDY 2040	0.944	0.874	0.908
CDY 2054	0.952	0.855	0.901
CDY 2046	0.952	0.849	0.898
CDY 2015	0.914	0.859	0.886
CFR 1638	0.928	0.885	0.906
CFR 1641	0.899	0.873	0.886
CFR 1639	0.930	0.841	0.883
CFR 1643	0.918	0.835	0.875
CFR 1666	0.951	0.807	0.873
CFR 1651	0.906	0.808	0.854
CSV 20180427 144535647	0.937	0.898	0.917
CSV 1877	0.928	0.879	0.903
CSV 20180427 144507419	0.855	0.867	0.861
CSV 1898	0.897	0.823	0.858
CSV 20180427 144723166	0.850	0.848	0.849
SVB 20180427 151818928	0.949	0.890	0.919
SVB 1954	0.912	0.915	0.913
SVB 1944	0.900	0.922	0.911
SVB 1935	0.926	0.856	0.889
SVB 1972	0.895	0.860	0.877
SYH 2017–04-27 1318	0.943	0.866	0.903
SYH 2017–04-27 1322	0.930	0.870	0.899
SYH 2017–04-27 1239	0.921	0.867	0.893
SYH 2017–04-27 1269	0.926	0.833	0.877
SYH 2017–04-27 1304	0.908	0.746	0.819
All pixels in test set	**0.920**	**0.860**	**0.889**

et al., 2010) are also presented. Considering the diversity in clusters sizes and shapes, IoU is specially important: higher values indicate better grape berries coverage, a desirable property for yield prediction applications. Lower IoU values indicate poorer berry coverage or disagreement in clusters segmentation between prediction and the ground truth, that means divergence in the berries' assignment to clusters.

Fig. 7 shows five examples of instance segmentation results produced by the Mask R-CNN. It illustrates the network capability to learn shape, compactness and color variability, and discriminate occluding foreground as branches and trunks. Inter-variety color variation (*Chardonnay/Sauvignon Blanc vs. Cabernet/Syrah*) and intra-variety color variation (*Syrah* and *Cabernet* maturity) are properly modeled by the network, as well as shape, size and elongation (*Chardonnay vs. Cabernet*, for example). The confidence level is also expressive: even considering that the confidence threshold is 0.9, most of the instances present levels equal or close to 1.0. Values lower than 0.99 can be observed in cases of severe occlusion, like the leftmost grape cluster in the *Syrah* example.

Grape cluster segmentation is challenging, even to the human annotators: occlusions and the absence of 3-D input or on-site annotation make the dataset error-prone regarding the correct segmentation of large agglomerations of clusters. Fig. 8 shows a case where segmentation divergence produces false negatives and false positives in the evaluation, although the almost correct detection of the grape berries. Note that for the clusters on the center in Fig. 8, the prediction looks like a more reasonable segmentation: two clusters are more plausible than one big and bifurcated cluster proposed by the ground-truth, but we would need in-field inspection (or a 3-D model) to check the correct detection. The other divergences in the figure illustrate the difficulty of the segmentation task: the clusters proposed by the prediction and the ground-truth look equally plausible, and again only an in-field checking could provide the right answer. Difficulties on successful cluster segmentation were also reported by Nuske et al. (2014). As noted before, segmentation divergence can also deteriorate IoU. Fortunately, we will see further in this work than data from a moving camera and 3-D association can relief such occlusion issues by properly integrating images from multiple poses.

R_{inst} and P_{inst} can suffer from erroneous segmentation, but what about semantic segmentation? As can be seen in Fig. 8, despite cluster segmentation errors, at the berry level most of the grape pixels look properly detected. To evaluate the detection of grape pixels, we use the measures R_{seman} and P_{seman}, recall and precision for the semantic segmentation variation of the problem. Table 4 shows the overall result for semantic segmentation on the entire masked set (last line), but also the results found for each one of the 27 images. The table groups the masked test set by the different varieties, allowing a comparison across different grape types. The overall F_1 score for semantic segmentation is 0.89 and no single variety has exhibited a remarkably different score. This is also an evidence that berries assignment to individual clusters (cluster segmentation) is the main factor affecting IoU.

Table 5 presents the results for object detection produced by the three networks, considering the entire test set of 837 clusters in 58 images. It is worth remembering that the models were trained using the *masked* training set, composed of 88 images (1,848 after augmentation), but the results in Table 5 show the evaluation for the entire "boxed" test set (considering intersection over union for the rectangular bounding boxes produced by Mask R-CNN). The recall values in the table show the YOLO networks lose more clusters if compared to the Mask R-CNN network, specially YOLOv3. Fig. 9 shows some examples of object detection for the three networks. In the figure we can see more false negatives (lost clusters) in the YOLOv3's results. Also in Table 5, as we require higher values of IoU (what means better adjusted bounding boxes), YOLO networks show worse results compared to Mask R-CNN.

To evaluate the spatial registration method and the potential of the entire methodology to address fruit counting, we employed a video sequence captured on field. Multiple counting is avoided by the tracking produced by the 3-D assignment: the number of individual tracks should correspond to the number of observed clusters in the video sequence, as seen previously in Fig. 6 (b). The video sequence was captured by a smartphone camera in full-HD (1, 920 × 1, 080 pixels) while a service vehicle moved along a row of vines. The keyframes of the MPEG video

Table 5
Object detection for all test set of WGISD: Mask R-CNN, YOLOv2 and YOLOv3.

	Mask R-CNN				YOLOv2				YOLOv3			
IoU	AP	P_{box}	R_{box}	F_1	AP	P_{box}	R_{box}	F_1	AP	P_{box}	R_{box}	F_1
0.300	0.805	0.907	0.873	0.890	0.675	0.893	0.728	0.802	0.566	0.901	0.597	0.718
0.400	0.777	0.891	0.858	0.874	0.585	0.818	0.667	0.735	0.494	0.829	0.550	0.661
0.500	0.719	0.856	0.824	0.840	0.478	0.726	0.591	0.652	0.394	0.726	0.481	0.579
0.600	0.611	0.788	0.759	0.773	0.288	0.559	0.455	0.502	0.261	0.587	0.389	0.468
0.700	0.487	0.697	0.671	0.684	0.139	0.390	0.318	0.350	0.125	0.405	0.269	0.323
0.800	0.276	0.521	0.502	0.511	0.027	0.172	0.140	0.154	0.036	0.205	0.136	0.164

Fig. 9. Some object detection results produced by the three neural networks: Mask R-CNN, YOLOv2 and YOLOv3, one example for each grape variety. Same color does not mean correspondence. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

sequence were extracted and the first 500 keyframes were employed in this evaluation. Employing keyframes from the MPEG stream is useful because (i) these frames present fewer compression artifacts than other frames in the video sequence, (ii) the number of images (frames) is reduced, and (iii) there is still sufficient overlap between frames to perform the feature correspondence needed by structure-from-motion and to provide multiple views for each grape cluster. Mask R-CNN inference was performed for each keyframe and the found mask stored. COLMAP was employed to create a sparse 3-D model by SfM. Finally, the spatial registration proposed on Section 3.5 was employed, matching the clusters along the frame sequence (Fig. 10). The results for the entire frame sequence can be seen in an available video.[5]

[5] https://youtu.be/1Hji3GS4mm4. Note the video is edited to a 4 frames/s rate to allow the viewer follow the tracks more easily.

5. Discussion

The Mask R-CNN network presented superior results as compared to the YOLO networks. Considering IoU values equal or superior to 0.5, the advantage of Mask R-CNN becomes more salient: even considering a 70% IoU, the F_1 score is impressive. As a reference, Sa et al. (2016) reported 0.828 and 0.848 F_1 scores for sweet peppers and rock melons respectively at 0.4 IoU using Faster R-CNN while Bargoti and Underwood (2017a) reported a 0.90 F_1 for apples and mangoes considering a 0.2 IoU, also employing Faster R-CNN. However, readers should keep in mind that it is just a reference, not a direct comparison or benchmark considering the different crops and datasets.

The use of three different scales by YOLOv3 could not be an advantage over YOLOv2 considering the almost constant distance between the camera and the vineyard row. In the same way, Mask R-CNN's use of FPN could be reconsidered. Agronomical constraints could

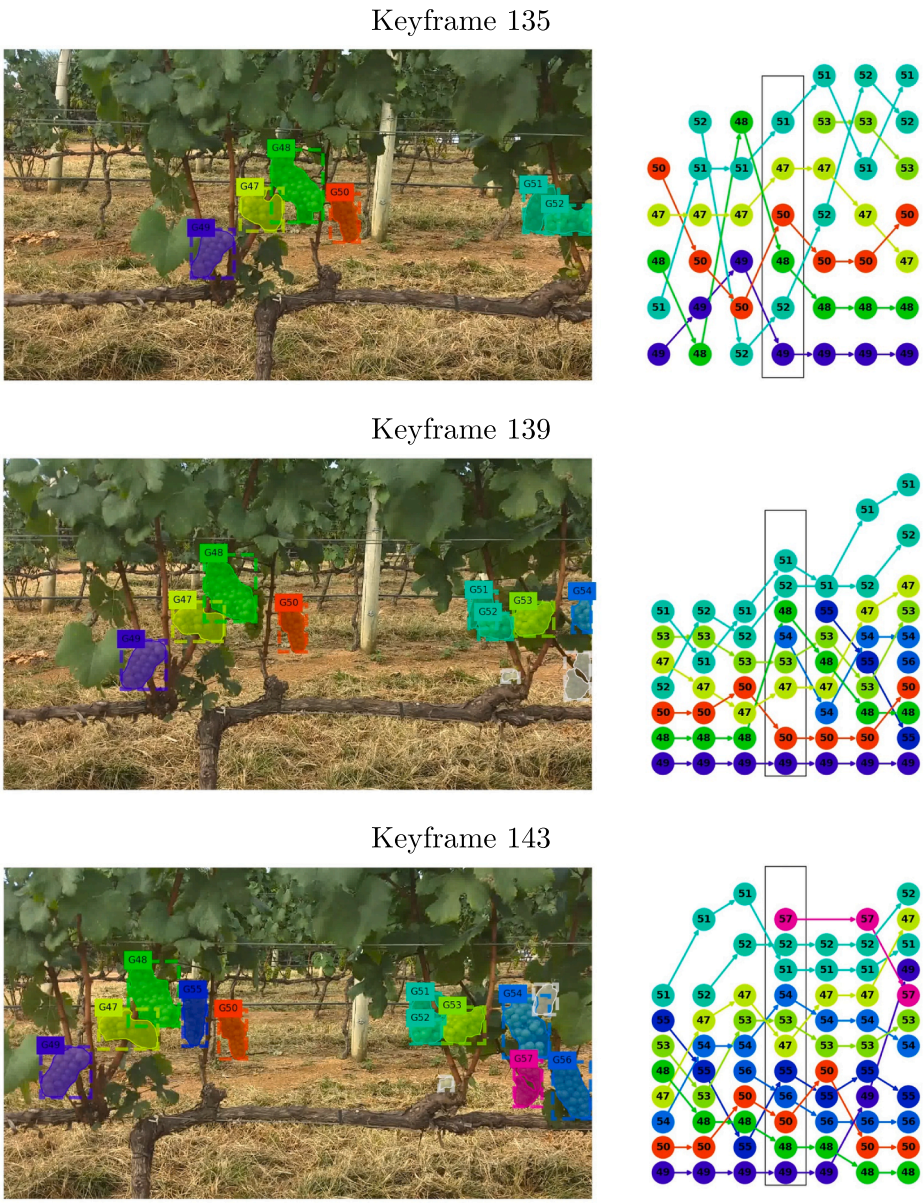

Fig. 10. Instance matching and tracking using 3-D assignment. (Left) Keyframes extracted from a video sequence by a 1080p camera. (Right) The graph-based tracking, similar to the one shown in Fig. 6. Colors and numbers on the keyframes correspond to the colors and number in the graph. See the available video for a more extensive demonstration. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

be explored: how big a group of berries should be to be considered a cluster? In other words, the operational and agronomical context should be explored to define the scales of interest. YOLOv3 employs multi-label classification, useful for problems presenting non-mutually exclusive object classes. However, considering our single class fruit detection problems, this would not be an advantage of YOLOv3 compared to YOLOv2. Considering that the YOLOv3 is deeper and, as consequence, prone to overfitting, it could need more data to reach and surpass the results of YOLOv2, as observed in Table 5. Although the better results produced by Mask R-CNN, it is important to note that rectangular bounding box annotation for object detection is faster to be produced, what means bigger datasets could be available to train networks like YOLOv2 and YOLOv3 – in this work, we have constrained YOLO training to the same dataset available to Mask R-CNN.

What could we be say about the *generalization* of these models? In other scenarios, like different developmental stages, crop production systems or camera poses, they would be able to detect grape clusters properly? Fig. 11 shows the results produced by the Mask R-CNN for images collected from the Internet[6]). No parameter tuning or other adaptions was employed. Fig. 11(a) and (b) show mature crops in camera poses very different from the images in WGISD, the latter showing a worker and a tool – elements that are not present in the dataset. The network was able to detect of a considerable part of the instances. Fig. 11(c) shows an example where false positives appeared in parts of textured leaves, but most of the berries were detected and the predicted clusters looks plausible. Finally, Fig. 11(d) shows clusters in an earlier developmental stage. Apparently, the network is splitting the clusters in regions of higher compactness, probably because the training

[6] The four images in Fig. 11 are public-domain or Creative Commons-licensed pictures. Their authors are McKay Savage (https://www.flickr.com/photos/56796376@N00/4034779039), Hahn Family Wines (https://commons.wikimedia.org/wiki/File:Hand_harvesting_wine_grape.jpg), Peakpx (http://www.peakpx.com/564748/green-grape-vine) and Circe Denyer (https://www.publicdomainpictures.net/en/view-image.php?image=298996.

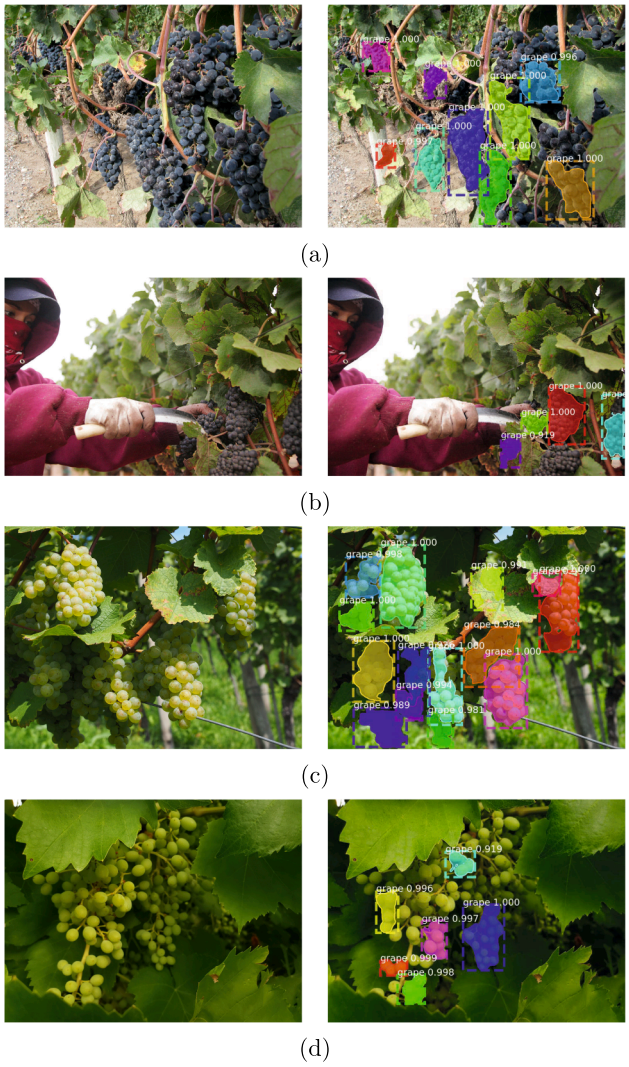

Fig. 11. Mask R-CNN generalization on novel scenarios without any tuning. (a)–(b) Examples presenting different camera poses. (c) Different pose and leaf texture. (d) Different developmental stage. These are Creative Commons-licensed images obtained from Internet (see text for references).

set in WGISD provides examples of more compact clusters. Such qualitative results indicate the model present good generalization and accurate results could be produced by tuning and transfer learning.

The presented CNN-based detectors can be integrated in larger systems that, employing a data association strategy, will be able to integrate the detections and perform localized fruit counting on site. A moving camera, combined to fruit tracking, is a powerful way to deal with occlusions due the integration of different camera poses. Integrating video information can also relieve localized errors in fruit detection in a few frames. As shown, an ordinary 1080p RGB camera can produce input for accurate results, being an affordable approach to fruit counting and orchard inspection. Such vision systems can be easily integrated in tractors, implements, service vehicles, robots and UAVs, possibly employing high performance processing units (GPUs and TPUs) with low energy consumption or even edge computing (Satyanarayanan, 2017).

Notwithstanding, while our spatial integration is employing a computational-intensive process such as structure-from-motion, other implementations could use SLAM algorithms (simultaneous localization and mapping), the real-time formulation of SfM. Liu et al. (2019) avoided the computationally-intensive process of feature detection and matching in SfM by employing the fruits' centers found by Faster R-CNN and Kalman Filter tracking for inter-frame association. In other words, the fruits became the *landmarks* for the SfM procedure (implemented in COLMAP). However, it is unclear what happens if *no fruits* are available in a segment of the video sequence. A fast SLAM algorithm such as ORB-SLAM (Mur-Artal and Tardos, 2017) or SVO (Forster et al., 2014), not relying on any specific landmark, could be a more robust alternative.

6. Conclusion

Computer vision's current maturity level can produce impressive and robust results in photogrammetry and perceptual tasks, even in challenging outdoor environments such as agricultural orchards. Combining structure-from-motion (or its real-time version: SLAM) and convolutional neural networks, advanced monitoring and robotics applications can be developed for agriculture and livestock.

This work presents a methodology for grape detection, tracking and counting in vineyards employing a single off-the-shelf 1080p camera. We have reached F_1 scores superior to 0.9 for instance detection in wine grapes, a challenging crop that presents enormous variability in shape, size, color and compactness. We also showed that 3-D models produced by structure-from-motion or SLAM can be employed to track fruits, avoiding double counts and increasing tolerance to errors in detection. The same methodology could be used successfully for other crops produced in trellis-like systems such as apples, peaches and berries. Adaptions of the methodology can be developed for fruits grown in trees presenting bigger canopies, like citrus and mangoes – yield could be estimated from regression from the visible fruit counts.

Further research could consider more integration between the photogrammetry and perception modules, looking for more sophisticated *scene understanding* systems able to robustly cope with occlusions and other sources of errors.

Declaration of Competing Interest

There are no conflict of interest in this work.

Acknowledgments

This work was supported by the Brazilian Agricultural Research Corporation (Embrapa) under grant 01.14.09.001.05.04 and by CNPq PIBIC Program (grants 161165/2017-6 and 125044/2018-6). S. Avila is partially funded by Google LARA 2018 & 2019, FAPESP (2017/16246-0, 2013/08293-7), and FAEPEX (3125/17). We thank Amy Tabb who provided helpful comments on the first version of the manuscript. We also thank to Luís H. Bassoi and Luciano V. Koenigkan for their generous support and help. Special thanks to Guaspari Winery for allowing image data collection.

Appendix A. Embrapa Wine Grape Instance Segmentation Dataset – Embrapa WGISD

This section presents a detailed description of the dataset, a *datasheet for the dataset* as proposed by Gebru et al. (2018).

A.1. Motivation for Dataset Creation

A.1.1. Why was the dataset created?

Embrapa WGISD (*Wine Grape Instance Segmentation Dataset*) was created to provide images and annotation to study *object detection and instance*

segmentation for image-based monitoring and field robotics in viticulture. It provides instances from five different grape varieties taken from the field. These instances show variance in grape pose, illumination and focus, including genetic and phenological variations such as shape, color and compactness.

A.1.2. What (other) tasks could the dataset be used for?

Possible uses include relaxations of the instance segmentation problem: classification (Is a grape in the image?), semantic segmentation (What are the "grape pixels" in the image?), and object detection (Where are the grapes in the image?). The WGISD can also be used in grape variety identification.

A.1.3. Who funded the creation of the dataset?

The building of the WGISD dataset was supported by the Embrapa SEG Project 01.14.09.001.05.04, *Image-based metrology for Precision Agriculture and Phenotyping*, and the CNPq PIBIC Program (grants 161165/2017–6 and 125044/2018–6).

A.2. Dataset Composition

A.2.1. What are the instances?

Each instance consists of an RGB image and an annotation describing grape clusters locations as bounding boxes. A subset of the instances also contains binary masks identifying the pixels belonging to each grape cluster. Each image presents at least one grape cluster. Some grape clusters can appear far at the background and should be ignored.

A.2.2. Are relationships between instances made explicit in the data?

File names prefixes identify the variety observed in the instance (Table 1).

A.2.3. How many instances of each type are there?

The dataset consists of 300 images containing 4,432 grape clusters identified by bounding boxes. A subset of 137 images also contains binary masks identifying the pixels of each cluster. It means that from the 4,432 clusters, 2,020 of them present binary masks for instance segmentation, as summarized in Table 1.

A.2.4. What data does each instance consist of?

Each instance contains an 8-bit RGB image and a text file containing one bounding box description per line. These text files follow the "YOLO format" (Redmon et al., 2016):

```
CLASS CX CY W H
```

class is an integer defining the object class – the dataset presents only the grape class that is numbered 0, so every line starts with this "class zero" indicator. The center of the bounding box is the point (c_x, c_y), represented as float values because this format normalizes the coordinates by the image dimensions. To get the absolute position, use $(2048 \cdot c_x, 1365 \cdot c_y)$. The bounding box dimensions are given by W and H, also normalized by the image size.

The instances presenting mask data for instance segmentation contain files presenting the `.npz` extension. These files are compressed archives for NumPy n-dimensional arrays (Van Der Walt et al., 2011). Each array is a $H \times W \times n_{\text{clusters}}$ three-dimensional array where n_{clusters} is the number of grape clusters observed in the image. After assigning the NumPy array to a variable `M`, the mask for the i-th grape cluster can be found in `M[:,:,i]`. The i-th mask corresponds to the i-th line in the bounding boxes file.

The dataset also includes the original image files, presenting the full original resolution. The normalized annotation for bounding boxes allows easy identification of clusters in the original images, but the mask data will need to be properly rescaled if users wish to work on the original full resolution.

A.2.5. Is everything included or does the data rely on external resources?

Everything is included in the dataset.

A.2.6. Are there recommended data splits or evaluation measures?

The dataset comes with specified train/test splits. The splits are found in lists stored as text files. There are also lists referring only to instances presenting binary masks.

Standard measures from the information retrieval and computer vision literature should be employed: precision and recall, F_1 score and average precision as seen in Lin et al. (2014) and Everingham et al. (2010).

A.2.7. What experiments were initially run on this dataset?

To the present date, this work describes the first experiments run on this dataset.

A.3. Data collection process

A.3.1. How was the data collected?

Images were captured at the vineyards of Guaspari Winery, located at Espírito Santo do Pinhal, São Paulo, Brazil (Lat −22.181018, Lon −46.741618). The winery staff performs dual pruning: one for shaping (after previous year harvest) and one for production, resulting in canopies of lower density. The image capture was realized in April 2017 for *Syrah* and in April 2018 for the other varieties (see Table 1). No pruning, defoliation or any intervention in the plants was performed specifically for the dataset construction: the images capture a real, trellis system-based wine grape production. The camera captures the images in a frontal pose, that means the camera principal axis is approximately perpendicular to the wires of the

trellis system and the plants rows.

A Canon™ EOS REBEL T3i DSLR camera and a Motorola™ Z2 Play smartphone were used to capture the images. The cameras were located between the vines lines, facing the vines at distances around 1–2 meters. The EOS REBEL T3i camera captured 240 images, including all *Syrah* pictures. The Z2 smartphone grabbed 60 images covering all varieties except *Syrah*. The REBEL images were scaled to 2048×1365 pixels and the Z2 images to 2048×1536 pixels (see Section A.4.1). More data about the capture process can be found in the Exif data found in the original image files, included in the dataset.

A.3.2. Who was involved in the data collection process?

The authors of this paper. T.T. Santos, A.A. Santos and S. Avila captured the images in field. T.T. Santos, L.L. de Souza and S. Avila performed the annotation.

A.3.3. How was the data associated with each instance acquired?

The rectangular bounding boxes identifying the grape clusters were annotated using the `labelImg` tool.[7] The clusters can be under severe occlusion by leaves, trunks or other clusters. Considering the absence of 3-D data and on-site annotation, the clusters locations had to be defined using only a single-view image, so some clusters could be incorrectly delimited.

A subset of the bounding boxes was selected for mask annotation, using a novel tool developed by the authors and presented in this work. This interactive tool lets the annotator mark grape and background pixels using scribbles, and a graph matching algorithm developed by Noma et al. (2012) is employed to perform image segmentation to every pixel in the bounding box, producing a binary mask representing grape/background classification.

A.4. Data preprocessing

A.4.1. What preprocessing/cleaning was done?

The following steps were taken to process the data:

1. Bounding boxes were annotated for each image using the `labelImg` tool.
2. Images were resized to $W = 2048$ pixels. This resolution proved to be practical for mask annotation, a convenient balance between grape detail and time spent by the graph-based segmentation algorithm.
3. A randomly selected subset of images was employed for mask annotation using the interactive tool based on graph matching.
4. All binaries masks were inspected, in search of pixels attributed to more than one grape cluster. The annotator assigned the disputed pixels to the most likely cluster.
5. The bounding boxes were fitted to the masks, which provided a fine-tuning of grape cluster locations.

A.4.2. Was the "raw" data saved in addition to the preprocessed data?

The original resolution images, containing the Exif data provided by the cameras, is available in the dataset.

A.5. Dataset distribution

A.5.1. How is the dataset distributed?

The dataset is available at Zenodo (Santos et al., 2019).

A.5.2. When will the dataset be released/first distributed?

The dataset was released in July 2019.

A.5.3. What license (if any) is it distributed under?

The data is released under Creative Commons BY-NC 4.0 (Attribution-NonCommercial 4.0 International license). There is a request to cite the corresponding paper if the dataset is used. For commercial use, contact the Embrapa Agricultural Informatics business office at `cnptia.-parcerias@embrapa.br`.

A.5.4. Are there any fees or access/export restrictions?

There are no fees or restrictions. For commercial use, contact Embrapa Agricultural Informatics business office at `cnptia.parcerias@em-brapa.br`.

A.6. Dataset maintenance

A.6.1. Who is supporting/hosting/maintaining the dataset?

The dataset is hosted at Embrapa Agricultural Informatics and all comments or requests can be sent to Thiago T. Santos at `thiago.santos@embrapa.br` (maintainer).

A.6.2. Will the dataset be updated?

There are no scheduled updates. In case of further updates, releases will be properly tagged at GitHub.

[7] https://github.com/tzutalin/labelImg.

A.6.3. If others want to extend/augment/build on this dataset, is there a mechanism for them to do so?

Contributors should contact the maintainer by e-mail.

A.6.4. No warranty

The maintainers and their institutions are *exempt from any liability, judicial or extrajudicial, for any losses or damages arising from the use of the data contained in the image database*.

Appendix B. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.compag.2020.105247.

References

Acuna, D., Ling, H., Kar, A., Fidler, S., 2018. Efficient interactive annotation of segmentation datasets with Polygon-RNN++. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Alahi, A., Ortiz, R., Vandergheynst, P., 2012. FREAK: Fast Retina Keypoint. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 510–517. <http://ieeexplore.ieee.org/document/6247715/>. doi: https://doi.org/10.1109/CVPR.2012.6247715.

Barbedo, J.G.A., 2019. Plant disease identification from individual lesions and spots using deep learning. Biosyst. Eng. 180, 96–107. https://doi.org/10.1016/J.BIOSYSTEMSENG.2019.02.002.

Bargoti, S., Underwood, J., 2017a. Deep fruit detection in orchards. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 3626–3633. https://doi.org/10.1109/ICRA.2017.7989417. arXiv:arXiv:1610.03677v2.

Bargoti, S., Underwood, J.P., 2017b. Image segmentation for fruit detection and yield estimation in Apple Orchards. J. Field Robot. 34, 1039–1060. https://doi.org/10.1002/rob.21699.

Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. 35, 1798–1828. https://doi.org/10.1109/TPAMI.2013.50.

Chen, S.W., Shivakumar, S.S., Dcunha, S., Das, J., Okon, E., Qu, C., Taylor, C.J., Kumar, V., 2017. Counting apples and oranges with deep learning: a data-driven approach. IEEE Robot. Autom. Lett. 2, 781–788. https://doi.org/10.1109/LRA.2017.2651944.

Chollet, F., 2017. Deep learning with Python. Manning Publications Company.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 248–255.

Duckett, T., Pearson, S., Blackmore, S., Grieve, B., 2018. Agricultural robotics: The future of robotic agriculture. CoRR, abs/1806.06762. http://arxiv.org/abs/1806.06762. arXiv:1806.06762.

Dunn, G.M., Martin, S.R., 2004. Yield prediction from digital image analysis: a technique with potential for vineyard assessments prior to harvest. Aust. J. Grape Wine Res. 10, 196–198. https://doi.org/10.1111/j.1755-0238.2004.tb00022.x.

Dutta, A., Gupta, A., Zissermann, A., 2016. VGG image annotator (VIA). <http://www.robots.ox.ac.uk/vgg/software/via/>. Version: 2.0.6, Accessed: April 23, 2019.

Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2010. The pascal visual object classes (VOC) challenge. Int. J. Comput. Vision 88, 303–338.

Forster, C., Pizzoli, M., Scaramuzza, D., 2014. Svo: fast semi-direct monocular visual odometry. In: 2014 IEEE international conference on robotics and automation (ICRA). IEEE, pp. 15–22.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J.W., Wallach, H.M., III, H.D., Crawford, K., 2018. Datasheets for datasets. CoRR, abs/1803.09010. http://arxiv.org/abs/1803.09010. arXiv:1803.09010.

Girshick, R., 2015. Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448.

Gongal, A., Amatya, S., Karkee, M., Zhang, Q., Lewis, K., 2015. Sensors and systems for fruit detection and localization: a review. Comput. Electron. Agric. 116, 8–19. https://doi.org/10.1016/j.compag.2015.05.021.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press. <http://www.deeplearningbook.org>.

Hartley, R., Zisserman, A., 2003. Multiple View Geometry in Computer Vision, second ed. Cambridge University Press, New York, NY, USA.

He, K., Gkioxari, G., Dollar, P., Girshick, R., 2017. Mask R-CNN. In: The IEEE International Conference on Computer Vision (ICCV).

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K., 2017. Speed/accuracy trade-offs for modern convolutional object detectors. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Jung, A., 2019. imgaug documentation. <https://imgaug.readthedocs.io>. Revision: cce07845, Accessed: June 23, 2019.

Kamilaris, A., Prenafeta-Boldú, F.X., 2018. Deep learning in agriculture: a survey. Comput. Electron. Agric. 147, 70–90. https://doi.org/10.1016/J.COMPAG.2018.02.016.

Kicherer, A., Herzog, K., Bendel, N., Klück, H.-C., Backhaus, A., Wieland, M., Rose, J.C., Klingbeil, L., Läbe, T., Hohl, C., Petry, W., Kuhlmann, H., Seiffert, U., Töpfer, R., 2017. Phenoliner: a new field phenotyping platform for grapevine research. Sensors 17. https://doi.org/10.3390/s17071625.

Kirkpatrick, K., 2019. Technologizing agriculture. Commun. ACM 62, 14–16. https://doi.org/10.1145/3297805.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems 25. Curran Associates Inc., pp. 1097–1105.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. https://doi.org/10.1038/nature14539.

Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L., 2014. Microsoft COCO: Common Objects in Context. In: Fleet, D, Pajdla, T., Schiele, B., Tuytelaars, T. (Eds.), Computer Vision – ECCV 2014. Springer International Publishing, pp. 740–755.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikäinen, M., 2018. Deep Learning for Generic Object Detection: A Survey, http://arxiv.org/abs/1809.02165. arXiv:1809.02165.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. Ssd: Single shot multibox detector. In: European conference on computer vision. Springer, pp. 21–37.

Liu, X., Chen, S.W., Liu, C., Shivakumar, S.S., Das, J., Taylor, C.J., Underwood, J., Kumar, V., 2019. Monocular camera based fruit counting and mapping with semantic data association. IEEE Robot. Autom. Lett. 4, 2296–2303. https://doi.org/10.1109/LRA.2019.2901987.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60, 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94.

Loy, G., Zelinsky, A., 2003. Fast radial symmetry for detecting points of interest. IEEE Trans. Pattern Anal. Mach. Intell. 25, 959–973. https://doi.org/10.1109/TPAMI.2003.1217601.

Matterport, Inc, 2018. Mask R-CNN for Object Detection and Segmentation. <https://github.com/matterport/Mask_RCNN>. Commit: 4f440de, Accessed: June 23, 2019.

Mur-Artal, R., Tardos, J.D., 2017. ORB-SLAM2: an Open-Source SLAM system for monocular, stereo, and RGB-D Cameras. IEEE Trans. Rob. 33, 1255–1262. https://doi.org/10.1109/TRO.2017.2705103. arXiv: 1610.06475.

Noma, A., Graciano, A.B., Cesar, R.M., Consularo, L.A., Bloch, I., 2012. Interactive image segmentation by matching attributed relational graphs. Pattern Recogn. 45, 1159–1179. https://doi.org/10.1016/j.patcog.2011.08.017.

Nuske, S., Achar, S., Bates, T., Narasimhan, S., Singh, S., 2011. Yield estimation in vineyards by visual grape detection. In: IEEE International Conference on Intelligent Robots and Systems, pp. 2352–2358. https://doi.org/10.1109/IROS.2011.6048830.

Nuske, S., Wilshusen, K., Achar, S., Yoder, L., Singh, S., 2014. Automated visual yield estimation in vineyards. J. Field Robot. 31, 837–860. https://doi.org/10.1002/rob.21541.

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.

Redmon, J., Farhadi, A., 2017. Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7263–7271.

Redmon, J., Farhadi, A., 2018. YOLO v3: An Incremental Improvement [DB]. arXiv preprint arXiv:1612.08242.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, vol. 28. Curran Associates Inc., pp. 91–99.

Rose, J., Kicherer, A., Wieland, M., Klingbeil, L., Töpfer, R., Kuhlmann, H., 2016. Towards automated large-scale 3D phenotyping of vineyards under field conditions. Sensors 16, 2136. https://doi.org/10.3390/s16122136.

Roser, M., 2019. Employment in agriculture. Our World in Data,. <https://ourworldindata.org/employment-in-agriculture>.

Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., McCool, C., 2016. DeepFruits: a fruit detection system using deep neural networks. Sensors 16, 1222. https://doi.org/10.3390/s16081222.

Santos, T., de Souza, L., dos Santos Andreza, Avila, S., 2019. [dataset] Embrapa Wine Grape Instance Segmentation Dataset – Embrapa WGISD. doi: 10.5281/zenodo.3361736.

Santos, T.T., Bassoi, L.H., Oldoni, H., Martins, R.L., 2017. Automatic grape bunch detection in vineyards based on affordable 3D phenotyping using a consumer webcam.

In: Barbedo, J.G.A., Moura, M.F., Romani, L.A.S., Santos, T.T., Drucker, D.P. (Eds.), Anais do XI Congresso Brasileiro de Agroinformática (SBIAgro 2017). Unicamp, Campinas, pp. 89–98. http://ainfo.cnptia.embrapa.br/digital/bitstream/item/169609/1/Automatic-grape-SBIAgro.pdf.

Satyanarayanan, M., 2017. The emergence of edge computing. Computer 50, 30–39.

Scaramuzza, D., Fraundorfer, F., 2011. Visual Odometry [Tutorial]. IEEE Robot. Autom. Magaz. 18, 80–92. https://doi.org/10.1109/MRA.2011.943233.

Schönberger, J.L., Frahm, J.-M., 2016. Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR).

Shelhamer, E., Long, J., Darrell, T., 2017. Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39, 640–651. https://doi.org/10.1109/TPAMI.2016.2572683.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In ICLR. URL http://arxiv.org/abs/1409.1556.

Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W., 2000. Bundle Adjustment — A Modern Synthesis Vision Algorithms: Theory and Practice. In: Triggs, B., Zisserman, A., Szeliski, R. (Eds.), Vision Algorithms: Theory and Practice book part (with own title) 21.. Springer, Berlin/Heidelberg volume 1883 of Lecture Notes in Computer Science, pp. 153–177. doi: 10.1007/3-540-44480-7_21.

Van Der Walt, S., Colbert, S.C., Varoquaux, G., 2011. The NumPy array: a structure for efficient numerical computation. Comput. Sci. Eng. 13, 22.

Vincent, L., Soille, P., 1991. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Trans. Pattern Anal. Mach. Intell. 583–598.