2023 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor) | 979-8-3503-1272-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/MetroAgriFor58484.2023.10424094

STEWIE: eSTimating grapE berries number and radius from images using a Weakly supervIsed nEural network

Davide Botturi
DIMI, University of Brescia
Brescia, Italy
0000-0003-1831-5994
davide.botturi@unibs.it

Alessandro Gnutti
DII, University of Brescia
Brescia, Italy
0000-0002-8308-0776
alessandro.gnutti@unibs.it

Cristina Nuzzi
DIMI, University of Brescia
Brescia, Italy
0000-0001-5530-6136
cristina.nuzzi@unibs.it

Bernardo Lanza
DIMI, University of Brescia
Brescia, Italy
0009-0005-3561-754X
bernardo.lanza@unibs.it

Simone Pasinetti
DIMI, University of Brescia
Brescia, Italy
0000-0002-5098-6395
simone.pasinetti@unibs.it

***Abstract*—Counting tasks with overlapping and occluded targets are often tackled by means of neural networks outputting density maps. While this approach has been proven to be highly effective for crowd-counting tasks, it has not been exploited extensively in other fields (like fruit counting). Furthermore, this approach has never been used to infer the shape or the size of the recognized objects. In this paper, we present a novel deep learning-based methodology to automatically estimate the number of grape berries present in an image and evaluate their average radius as a double output of the network. For the model training, we employ a public dataset consisting of 300 vines images, where each berry center has been dot-annotated. Since the dataset does not directly provide information about the berry radii, we first develop a numerical optimization methodology to calculate the radius of the berries, by exploiting the dot annotations, some prior knowledge (berry maximum size), and a current state-of-the-art segmentation model. Then, we employ the combined information (berry center and radius) to train a custom neural network that outputs two density maps, from which we infer the number of berries in the image and their average size.**

***Index Terms*—Measurement science, grapevine berry, fruit counting, fruit size estimation, fruit yield estimation, viticulture, neural network, deep learning, density maps**

I. Introduction

Grapes are one of the world's most valuable crops, with a growing market including several food products such as fresh table fruit, raisins, wine, distillates, and juice concentrate. Several species of cultivar exist, mainly differentiated between white, red, and black varieties. Berries may be big or tiny, round or elongated according to species [1]. According to the final product they are being cultivated for, grapes may be grown using different approaches aimed at maximizing a specific characteristic, e.g., the sugar content or the berry size and color. Even the harvesting method is different since the preservation of the berry is of utmost importance in the case of table fruit and raisins production. Instead, it is not so relevant for wine and distillate production, since the berries will be smashed anyway after harvesting to extract the juice. However, this difference adds complexity to the harvesting task since automated berry-picking devices need to carefully detect the grape clusters and pick them up without damaging them [2].

Relevant investing is being done worldwide toward robotic and fully automated systems for fruit picking, fertilization, and crop harvesting to face the problem of the increasing food demand [3]. Alongside the mechanical design of the moving machine, the research community focused its attention on detection strategies and algorithms to equip robots with the necessary intelligence, leveraging knowledge already being used by production companies [4]. A fundamental topic in viticulture research is the estimation of yield production, which is important for the economic management of crop fields [5]. Currently, agronomists estimate the field's yield production by means of manual measurements, considering (i) the number of vines, (ii) the number of grape clusters per vine, and (iii) the number of berries per cluster, all combined to estimate the overall weight of grown fruit and the productivity of the field [6], [7]. Finally, in the context of Precision Agriculture (PA), tasks such as phenotyping, crop health monitoring, and precise localization of fruits [8], [9] even at early stages of maturation are of utmost interest to researchers and food production companies [10]. What all those applications have in common is the necessity to detect and localize in space both the grape clusters and the individual berries belonging to them. To this aim, contactless sensors are the most promising and effective devices that could be adopted in such complex and unstructured environments. Moreover, the recent advances in computer vision (CV) and artificial intelligence (AI) research gave way to a plethora of applications that were unthinkable

979-8-3503-1272-0/23/$31.00 ©2023 IEEE

just a decade ago, extensively exploiting vision systems such as color cameras and beyond visible light sensors such as near-infrared (NIR), thermal, and hyperspectral cameras.

As stated by [7], robust yield prediction by means of vision-based approaches is achieved by improving the accuracy of the single berries counting in images. Moreover, by computing the average size of the berries, it is possible to estimate the total weight of fruits seen by the camera. Despite being published in 2011, the work in [7] tackled the problem by fusing the color information with the depth information, hence segmenting efficiently the grape clusters and the single berries by means of CV image processing techniques. Similar works adopted a combination of 2D and 3D data for this purpose [11], [10], [4]. However, relying on 3D measurements in the field can be tricky, not to mention that AI models that directly take as input 3D information are not so common, because they typically require a lot of computational power and result in low-speed inference. Hence, they are currently not suitable for fast and robust in-field analysis. Therefore, it is not surprising that the research community mostly focused on 2D image analysis, also considering the astonishing performance of AI models for this type of visual data. For example, Neural Networks have been adopted in [12] for the automatic segmentation of grape berries using Deep Learning (DL). Several works also analyzed the most relevant parameters to tune DL models, such as the color space of the input images, the model architecture, and the impact of different augmentation techniques [13], [2]. Such studies are of utmost importance to develop a robust algorithm able to generalize between grape varieties, because the detection of white berries is more complex compared to red and black berries due to their distinctive color. A very promising approach to counting single berries in the clusters has been detailed in [14], where the authors adopted a custom algorithm that uses the berries' edge contour, the concave points, and their curvature to guide the counting. The idea of leveraging the berry edge to improve the berries segmentation is expanded in [15], where authors defined the berries' edge as a new segmentation class alongside the whole berry and the background classes.

However, among these works, no one tried to output an average estimation of the berries' size directly from the AI model. The only two that tried to also produce a measure of their radius were doing so by means of image processing and geometrical or morphological methods after the model produced the segmentation mask [14], [15]. In contrast, this work focuses on the estimation of the total number of berries in the image and of their average radius automatically at the same time, without the need to write complex algorithms to analyze the shape of the output segmentation mask.

To do so, we implement a customized Neural Network capable of generating two density maps: one for estimating the number of berries and another for estimating their radii. For the estimation of the berry count, traditional methods typically transform the dot annotations into a ground-truth density map using a Gaussian kernel. However, determining the optimal kernel size can be challenging and may lead to inaccurate results. Thus, inspired by [16], we employ a Bayesian loss function to exploit the dot annotations for learning a more accurate density map. Regarding the estimation of berry radii, since the dataset does not directly provide this information, we first develop a numerical optimization methodology to calculate the average radius of the berries, by leveraging the dot annotations, prior knowledge (such as the maximum size of berries), and a modified version of the well-known Segment Anything Model [17]. Then, by combining the information of the berry centers and radii, we can learn an accurate density map specifically for the mean radius estimation.

II. Proposed Methodology

The purpose of this work is to develop an algorithm to extract the number of berries and their average radius in pixels from an image. The algorithm consists of a neural network adapted from [16], which takes in input one image and outputs two density maps, called D^n and D^r, used to predict the estimated number $\tilde{N}$ of berries and their estimated average radius $\tilde{r}_{\text{mean}}$, respectively (see Fig. 1). The architecture of the proposed neural network is discussed in Section II-A.

Our model has been trained and tested on Embrapa WGISD [18], a public dataset designed for object detection and instance segmentation in viticulture. The dataset, described comprehensively in Section II-B, contains dot annotations that approximate the center positions of the berries. These dot annotations play a crucial role in training our model to learn the density maps D^n and D^r. In particular, drawing inspiration from the approach presented in [16], we employ the dot annotations to generate a likelihood map that represents the probability of a pixel belonging to a specific berry. These probabilities are then leveraged to learn the density map D^n. Instead, the likelihood map alone is insufficient for estimating D^r due to the lack of information regarding the berry radii

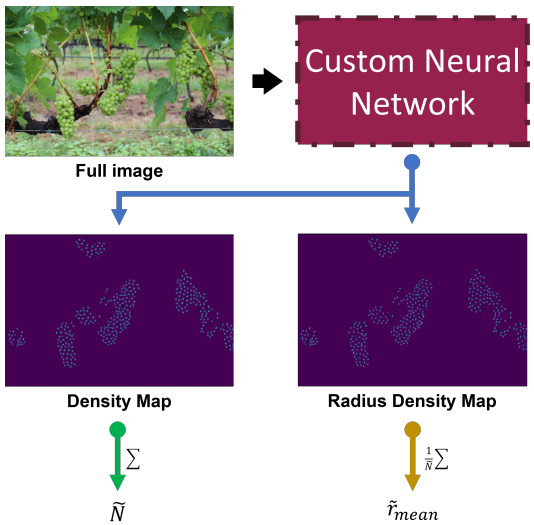

Fig. 1. Scheme of the inference process. The image is elaborated by the Custom Neural Network and two probability density maps are returned as output. Pixel densities are summed to compute the estimate of the number of berries $\tilde{N}$ and their average size $\tilde{r}_{\text{mean}}$.

in the dataset. To address this limitation, we integrate a numerical optimization framework that allows us to estimate the radii of the berries, which will be subsequently used in the derivation of D^r. A detailed discussion on the construction of the likelihood map and the numeric estimation of the berry radii is provided in Section II-C, which covers the details of the training process.

A. Custom Neural Network Architecture

The proposed neural net takes in input an image with size $H \times W$ and outputs two density maps D^n and D^r, used to esteem the number of berries and the mean radius, respectively, both with size $H \times W$. The architecture utilizes VGG-19 [19] as a backbone, which is a standard classification network. The backbone is followed by a bilinear interpolation layer which scales the backbone's output to $1/8$ of the size of the input image. Subsequently, a regression header is incorporated into the architecture, comprising two 3×3 convolutional layers with 256 and 128 channels, respectively, along with two 1×1 convolutional layers. This regression header generates the two density maps, which are then appropriately scaled using an interpolation layer to match the size of the input image. Our network mainly differs from the one described in [16] by incorporating two 1×1 convolutional layers instead of a single layer. This modification is crucial as our network is designed to output not only the density map used for estimating the number of berries, but also the density map required for the estimation of berry radii.

B. Dataset

The Embrapa WGISD includes 300 images of grape clusters from five different grape varieties (Chardonnay, Cabernet Franc, Cabernet Sauvignon, Sauvignon Blanc, and Syrah), with variations in pose, illumination, and focus, as well as genetic and phenological differences. To capture the images, a Canon EOS REBEL T3i DSLR camera, and a Motorola Z2 Play smartphone were used. The cameras were positioned between the vine lines at distances of 1-2 meters, with the EOS REBEL T3i camera capturing 240 images, including all Syrah pictures, and the Z2 smartphone taking 60 images of all other grape varieties. The resulting images were scaled to 2048×1365 pixels for the REBEL and 2048×1536 pixels for the Z2. Additional details about the image capture process can be found in the Exif data of the original image files, which are included in the dataset. In all 300 images, Geng Deng and colleagues [20] provided point-based annotations identifying a total of 187,374 berries.

C. Training process

The proposed algorithm has been trained on a random selection of 225 images, which accounts for 75% of the total 300 available images. The training procedure is repeated 600 times (training epochs) and once for each image of the training set (batch size equal to 1). The training was performed employing Pytorch 1.14, a framework for Tensors and Dynamic neural networks in Python with strong GPU acceleration [21].

The overall training process is illustrated in Fig. 2. The diagram showcases the different blocks involved, and their descriptions are provided below.

1) Data Loading and Preparation: First, a data loader picks an image (that has not been picked yet during the current epoch) and its corresponding dot annotations from the train set. Then, a random crop with a 512×512 size is extracted from the image. The cropped image with the associated dot annotations is retained and passed to the next step.

2) Likelihood Map and Target Radius Numerical Computation: This block focuses on computing the likelihood

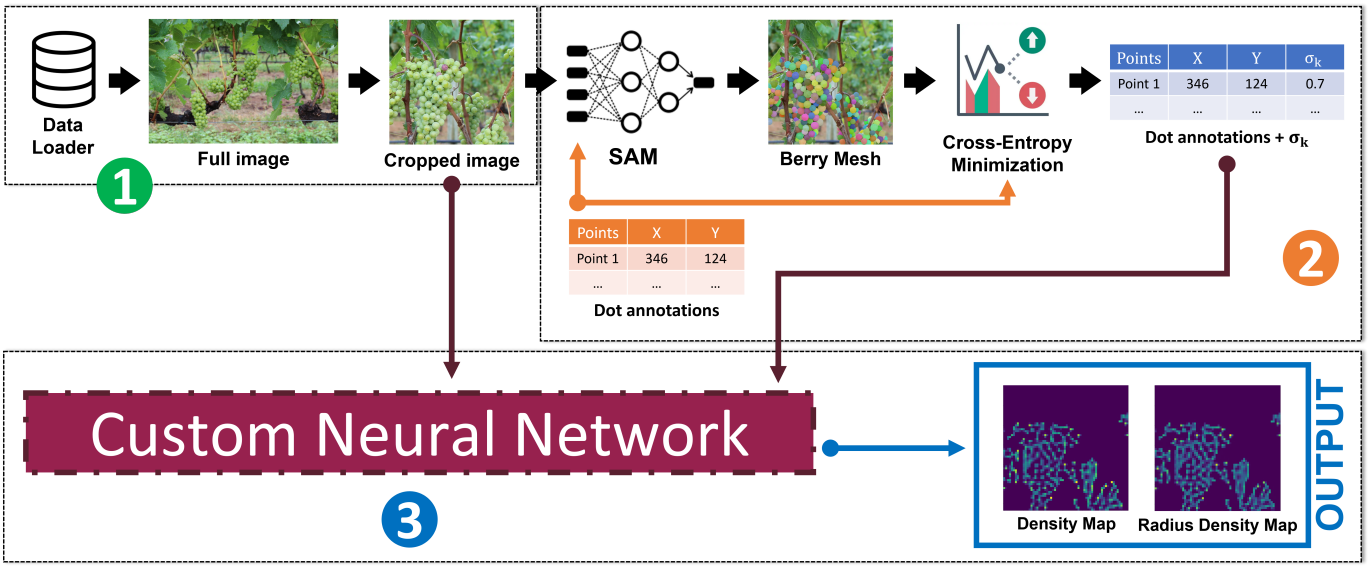

Fig. 2. Scheme of the training process. The procedure can be divided into three main blocks: (i) data loading and preparation, (ii) likelihood map computation and target radius estimation, and (iii) Bayesian loss and output generation.

map, which determines the probability of a pixel belonging to a specific berry, and numerically estimating the radius of the berries. These calculations are crucial for the subsequent calculation of the Bayesian loss used as function loss in training. To begin this process, the initial step is to segment all the berries within the image. This segmentation task is accomplished using the Segment Anything Model (SAM) [17], a cutting-edge instance segmentation model that utilizes the Mask-RCNN architecture to achieve accurate object detection and segmentation in images.

In our approach, we input the cropped image and corresponding dot annotations into SAM, which outputs a matrix of logits (M) with dimensions $512 \times 512 \times (N+1)$, where N represents the number of berries in the cropped image. We noted that sometimes SAM, when prompted with berry annotated dots, includes in the outputted mask pixels belonging to leaves or branches close to the selected berry. This happens more often when the target berries are heavily occluded (by leaves, brunches or other berries). In order to limit the impact of this erroneous behaviour we adopted the following strategy. While in the original SAM framework [17], M is directly processed to generate the final segmentation, we propose an additional processing step for M. In particular, we take into consideration the maximum berry size, manually identified as $r_{\text{max}} = 40$ pixels. To enforce this constraint, we generate a new logits matrix, denoted as M', as follows:

$$M'_{i,j,k} = \begin{cases} M_{i,j,k}, & \text{if } d_{i,j,k} \leq r_{\text{max}} \\ -\infty, & \text{otherwise} \end{cases} \tag{1}$$

where $d_{i,j,k}$ is the euclidean distance from the (i,j)-th pixel ($x_{i,j}$) and the k-th annotated berry (y_k). Then, M' is used to obtain the final segmentation mask, denoted as M_{seg}. This mask is a matrix of size $512 \times 512 \times (N+1)$, where the (i,j,k)-th element is assigned a value of 1 if $x_{i,j}$ belongs to y_k, otherwise, if $x_{i,j}$ does not belong to y_k, that element is set to 0.

At this stage, we can numerically compute the likelihood map P and the radii of the berries r_k by designing a new logits matrix M'' that enables the convergence of P to the segmentation mask M_{seg} through cross-entropy minimization. P will be a $512 \times 512 \times (N+1)$ matrix, where the (i,j,k)-th element, denoted as $P(x_{i,j} \in y_k)$, represents the probability of pixel $x_{i,j}$ belonging to the k_{th} berry (y_k) or the background (y_{N+1}). The computation of P is determined by the SoftMax function applied to M'', as follows:

$$P(x_{i,j} \in y_k) = \frac{e^{M''_{i,j,k}}}{\sum_{k=1}^{N+1} e^{M''_{i,j,k}}} \tag{2}$$

In our methodology, we design the matrix M'' as:

$$\begin{cases} M''_{i,j,k} = \frac{-d^2_{i,j,k}}{\sigma^2_k}, & \text{if } k \leq N \\ M''_{i,j,k} = -\sigma^2_{\text{ratio}}, & \text{if } k = N+1 \end{cases} \tag{3}$$

Here, σ_k is the optimization variable which controls the probability of $x_{i,j}$ belonging to y_k, and σ_{ratio} is a pre-defined constant that establishes the relationship between σ_k and the corresponding radius r_k. Specifically, $\sigma_{\text{ratio}} = \frac{r_k}{\sigma_k}$, independently of k. Basically, the magnitudes of $|M''_{i,j,k}|$ express the normalized distance from $x_{i,j}$ to y_k. The sigma values are initialized as $\sigma_k = \frac{r_{\text{max}}}{2 \cdot \sigma_{\text{ratio}}}$. These values are refined by minimizing the cross-entropy between P and M_{seg}. Upon optimization, the radius r_k of the k_{th} annotated berry is computed as $r_k = \sigma_k \cdot \sigma_{\text{ratio}}$, and the mean radius is given by $r_{\text{mean}} = \frac{1}{N} \sum_{k=1}^{N} r_k$. For clarity, we remark that $|M''_{i,j,N+1}|$ represents the normalized distance from $x_{i,j}$ to the background, which remains constant for all pixels. This design ensures that a pixel located on the edge of the k_{th} berry is equidistant from the k_{th} annotated dot (approximating the berry center) and the background. Specifically, when $d_{i,j,k} = r_k$, we have $M''_{i,j,k} = \frac{-r_k^2}{\sigma_k^2} = -\sigma^2_{\text{ratio}}$.

3) Bayesian Loss and output generation: To train the Neural Network to generate the desired density maps D^n and D^r, we employ the following custom Bayesian loss:

$$L^{Bayes} = L^n + L^r \tag{4}$$

L^n is the loss associated with the berry number estimation and L^r is the loss related to the berry mean radius. The two components of the loss are defined as:

$$\begin{aligned} L^n &= \sum_{k=1}^{N} |1 - E^n_k| + |E^n_{N+1}| \\ L^r &= \sum_{k=1}^{N} |\sigma_k - E^r_k| + |E^r_{N+1}| \end{aligned} \tag{5}$$

where E^n_k and E^r_k are computed as:

$$\begin{aligned} E^n_k &= \sum_{i=1}^{512} \sum_{j=1}^{512} D^n_{i,j} \cdot P(x_{i,j} \in y_k) \\ E^r_k &= \sum_{i=1}^{512} \sum_{j=1}^{512} D^r_{i,j} \cdot P(x_{i,j} \in y_k) \end{aligned} \tag{6}$$

Basically, L^n requires that for each of the annotated berry, the sum of the product between D^n and P is equal to one. Similarly, L^r requires that for each of the annotated berry, the sum of the product between D^r and P is equal to each berry's radius divided by σ_{ratio}. The subscript $N+1$ indicates the background, so adding E^n_{N+1} (E^r_{N+1}) to L^n (L^r) constraints D^n (D^r) to be as close as possible to 0 in those pixels belonging to the background.

At the inference stage, neither the dot annotations nor σ_k values are available, as well as the likelihood map P. Nethertheless, the two quantities of interest, namely, the total number N of berries and the mean radius r_{mean} of the berries, can be estimated as:

$$\begin{aligned} \tilde{N} &= \sum_{k=1}^{N+1} E^n_k = \sum_{k=1}^{N+1} \sum_{i=1}^{512} \sum_{j=1}^{512} D^n_{i,j} \cdot P(x_{i,j} \in y_k) = \\ &= \sum_{i=1}^{512} \sum_{j=1}^{512} D^n_{i,j} \cdot \sum_{k=1}^{N+1} P(x_{i,j} \in y_k) = \sum_{i=1}^{512} \sum_{j=1}^{512} D^n_{i,j} \end{aligned} \tag{7}$$

and

$$\begin{aligned}\tilde{r}_{\text{mean}} &= \frac{\sigma_{\text{ratio}}}{\tilde{N}} \sum_{k=1}^{N+1} E_r^n = \\ &= \frac{\sigma_{\text{ratio}}}{\tilde{N}} \sum_{k=1}^{N+1} \sum_{i=1}^{512} \sum_{j=1}^{512} D_{i,j}^r \cdot P(x_{i,j} \in y_k) = \\ &= \frac{\sigma_{\text{ratio}}}{\tilde{N}} \sum_{i=1}^{512} \sum_{j=1}^{512} D_{i,j}^r \cdot \sum_{k=1}^{N+1} P(x_{i,j} \in y_k) = \\ &= \frac{\sigma_{\text{ratio}}}{\tilde{N}} \sum_{i=1}^{512} \sum_{j=1}^{512} D_{i,j}^r\end{aligned} \tag{8}$$

In conclusion, the total number $\tilde{N}$ of berries can be determined by summing all the values of the D^n, while the mean radius $\tilde{r}_{\text{mean}}$ of the berries can be obtained by multiplying the sum of all the values of D^r with the pre-defined σ_{ratio} value and dividing it with the esteemed number of berries.

III. Experimental results

During the evaluation phase, our model has been tested on the unseen test set consisting of $K = 75$ images. Unlike the training phase, where images were cropped due to GPU limitations, the entire image is fed into the algorithm for testing. To assess the accuracy of the counting task, we utilize three metrics: mean absolute error (MAE), mean squared error (MSE), and mean absolute percentage error (MAPE). These metrics are calculated as follows:

$$\begin{aligned}\text{MAE} &= \frac{1}{K} \sum_{k=1}^{K} |N_k - \tilde{N}_k| \\ \text{RMSE} &= \frac{1}{K} \sqrt{\sum_{k=1}^{K} (N_k - \tilde{N}_k)^2} \\ \text{MAPE} &= \frac{100}{K} \sum_{k=1}^{K} \frac{|N_k - \tilde{N}_k|}{N_k}\end{aligned} \tag{9}$$

where N_k and $\tilde{N}_k$ are the ground-truth count and the estimated count for the k_{th} image, respectively.

While the ground-truth for the number of berries can be accurately obtained from the dot annotations, the ground-truth for the mean berry radius is not available initially. In order to assess the accuracy of our predictions regarding the berry radii, we calculate the mean absolute error (MAE_r) between the estimated mean berry radius ($\tilde{r}_{\text{mean}}$) to the reference mean radius (r_{mean}) computed as discussed in Section II-C2. However, it is important to acknowledge that r_{mean} is obtained from a process with unknown uncertainty. Therefore, in future developments, it will be necessary to validate the entire measurement chain to ensure the accuracy and reliability of our results.

Table I reports the results obtained on the test images for different values of σ_{ratio}. The proposed algorithm achieves remarkable performances in estimating the number of berries, with a low percentage error ranging from 3.7% to 7%. For the radius estimation, the algorithm still achieves interesting results, with an average error of approximately 2 pixels ($\tilde{r}_{\text{mean}}$ ranging from 15 to 35 pixels). Further enhancements could potentially be achieved by exploring and testing additional values of σ_{ratio} through an hyper parameter optimization. In this work we empirically identified $\sigma_{ratio} = 2$ as the best neural network among the tested one (lowest MAE and MAE_r values).

TABLE I
Experimental results using different values of σ_{ratio}.

σ_{ratio}	**MAE**	**RMSE**	**MAPE**	$\mathbf{MAE_r}$ [px]	$\mathbf{MAPE_r}$
0.75	37.7	49.7	6.3%	2.3	9.6%
1.00	40.2	50.1	6.7%	2.1	8.8%
2.00	21.7	31.0	3.6%	1.7	7.1%
3.00	22.2	30.0	3.7%	1.8	7.5%

In table II we show the results obtained by two other works for in-field berry counting compared to our best performing net($\sigma_{ratio} = 2$). Note that for [22] neither the $MAPE$ nor the average number of berries per image were reported. Since it has been validated on our same data-set (WGISD) we approximated the number of berries per image by dividing the total number of berries in the data set per the number of images (it might be different due to different train-test splits).

TABLE II
Comparisons with other works

Method	Avg. berry-number per image	**MAE**	**MAPE**
SAGBCNet [22]	$\simeq 624$	35.6	$\simeq 5.7\%$
GBCNET [23]	1093	117.4	10.7%
STEWIE	601	21.7	3.6%

IV. Conclusions

The novel methodology presented in this paper is able to robustly estimate the number of grape berries in a real in-field vineyard image, where several grape clusters are present as well as background noise such as leaves and branches, resulting in an overall estimation error less than 4%. In addition to the counting task, our proposed model also estimates the average radius of the berries, which is a novel contribution to AI-driven viticulture research. Our approach can be also applied to other fruits as long as the fruit has a spherical shape.

Despite being a promising work, some careful evaluation of the target radius estimation should be carried out in the future to (i) improve the current model by conducting a validation campaign of the radius estimation task, (ii) expand the model to accurately estimate the overall size of elongated and thin berries which have not a perfect spherical shape and (iii) address the conversion px to mm for fruit size estimate.

REFERENCES

[1] J. M. Alston and O. Sambucci, "Grapes in the world economy," in *The Grape Genome*, D. Cantu and M. A. Walker, Eds. Cham: Springer International Publishing, 2019, pp. 1–24.

[2] Y. Peng, A. Wang *et al.*, "A comparative study of semantic segmentation models for identification of grape with different varieties," *Agriculture*, vol. 11, no. 10, p. 997, Oct 2021.

[3] N. Biswas and A. Aslekar, "Improving agricultural productivity: Use of automation and robotics," in *2022 International Conference on Decision Aid Sciences and Applications (DASA)*, 2022, pp. 1098–1104.

[4] L. Luo, W. Yin *et al.*, "In-field pose estimation of grape clusters with combined point cloud segmentation and geometric analysis," *Computers and Electronics in Agriculture*, vol. 200, p. 107197, 2022.

[5] A. Barriguinha, M. de Castro Neto, and A. Gil, "Vineyard yield estimation, prediction, and forecasting: A systematic literature review," *Agronomy*, vol. 11, no. 9, p. 1789, Sep 2021.

[6] B. Komm and M. Moyer, *Vineyard yield estimation*. Washington State University Extension, 2015.

[7] S. Nuske, S. Achar *et al.*, "Yield estimation in vineyards by visual grape detection," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 2352–2358.

[8] R. Roscher, K. Herzog *et al.*, "Automated image analysis framework for high-throughput determination of grapevine berry sizes using conditional random fields," *Computers and Electronics in Agriculture*, vol. 100, pp. 148–158, 2014.

[9] M. Ferrer-Ferrer, J. Ruiz-Hidalgo *et al.*, "Simultaneous fruit detection and size estimation using multitask deep neural networks," *Biosystems Engineering*, vol. 233, pp. 63–75, 2023.

[10] T. T. Santos, L. L. de Souza *et al.*, "Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association," *Computers and Electronics in Agriculture*, vol. 170, p. 105247, 2020.

[11] Y. Peng, S. Zhao, and J. Liu, "Segmentation of overlapping grape clusters based on the depth region growing method," *Electronics*, vol. 10, no. 22, p. 2813, Nov 2021.

[12] R. Marani, A. Milella *et al.*, "Deep neural networks for grape bunch segmentation in natural images from a consumer-grade camera," *Precision Agriculture*, vol. 22, no. 2, pp. 387–413, Apr 2021.

[13] H. Cecotti, A. Rivera *et al.*, "Grape detection with convolutional neural networks," *Expert Systems with Applications*, vol. 159, p. 113588, 2020.

[14] L. Luo, W. Liu *et al.*, "Grape berry detection and size measurement based on edge image processing and geometric morphology," *Machines*, vol. 9, no. 10, p. 233, Oct 2021.

[15] L. Zabawa, A. Kicherer *et al.*, "Counting of grapevine berries in images via semantic segmentation using convolutional neural networks," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 164, pp. 73–83, 2020.

[16] Z. Ma, X. Wei *et al.*, "Bayesian loss for crowd count estimation with point supervision," 2019, arXiv preprint, arXiv:1908.03684.

[17] A. Kirillov, E. Mintun *et al.*, "Segment anything," 2023, arXiv preprint, arXiv:2304.02643.

[18] S. Thiago, L. de Souza *et al.*, "Embrapa Wine Grape Instance Segmentation Dataset – Embrapa WGISD," 2019, Online dataset, Zenodo.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, arXiv preprint, arXiv:1409.1556.

[20] G. Deng, T. Geng *et al.*, "TSGYE: Two-stage grape yield estimation," in *Neural Information Processing*, H. Yang, K. Pasupa *et al.*, Eds. Cham: Springer International Publishing, 2020, pp. 580–588.

[21] A. Paszke, S. Gross *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[22] Y. Tang, Y. Li *et al.*, "Field grape berry counting algorithm based on spatial attention mechanism," in *2022 China Automation Congress (CAC)*, 2022, pp. 2211–2216.

[23] L. Coviello, M. Cristoforetti *et al.*, "GBCNet: In-Field Grape Berries Counting for Yield Estimation by Dilated CNNs," *Applied Sciences*, vol. 10, p. 4870, 07 2020.