Simultaneous Fruit Detection and Size Estimation Using Multitask Deep Neural Networks

Mar Ferrer-Ferrer[a], Javier Ruiz-Hidalgo[b], Eduard Gregorio[a], Verónica Vilaplana[b], Josep-Ramon Morros[b], Jordi Gené-Mola[a,c,*]

[a]Research Group in AgroICT& Precision Agriculture - GRAP, Department of Agricultural and Forest Engineering, Universitat de Lleida (UdL) – Agrotecnio-CERCA Center, Lleida, Catalonia, Spain

[b]Department of Signal Theory and Communications, UniversitatPolitècnica de Catalunya, Barcelona, Catalonia, Spain.

[c]Efficient Use of Water in Agriculture Program, Institute of AgriFood, Research and Technology (IRTA), Parc Científic i Tecnològic Agroalimentari de Gardeny (PCiTAL), Fruitcentre, 25003 Lleida, Catalonia, Spain.

Abstract

The measurement of fruit size is of great interest to estimate the yield and predict the harvest resources in advance. This work proposes a novel technique for in-field apple detection and measurement based on Deep Neural Networks. The proposed framework was trained with RGB-D data and consists of an end-to-end multitask Deep Neural Network architecture specifically designed to perform the following tasks: 1) detection and segmentation of each fruit from its surroundings; 2) estimation of the diameter of each detected fruit. The methodology was tested with a total of 15335 annotated apples at different growth stages, with diameters varying from 27 mm to 95 mm. Fruit detection results reported an F1-score for apple detection of 0.88 and a mean absolute error of diameter estimation of 5.64 mm. These are state-of-the-art results with the additional advantages of: a) using an end-to-end multitask trainable network; b) an efficient and fast inference speed; and c) being based on RGB-D data which can be acquired with affordable depth cameras. On the contrary, the main disadvantage is the need of annotating a large amount of data with fruit masks and diameter ground truth to train the model. Finally, a fruit visibility analysis showed an improvement in the prediction when limiting the measurement to apples above 65% of visibility (mean absolute error of 5.09 mm). This suggests that future works should develop a method for automatically identifying the most visible apples and discard the prediction of highly occluded fruits.

Keywords: Fruit measurement; Yield estimation; Fruit visibility; Deep Learning; Precision Agriculture

1. Introduction

According to the Food and Agriculture Organization (FAO), by 2050 the agriculture industry will need to produce 70% more food while only being able to use 5% more land. Since most land suitable for farming is already in use, this production growth has to come from another source. The introduction of Precision Agriculture has enabled farmers to measure, map and manage crops at their different stages to increase production while optimising used resources and costs.

* *Corresponding author.*

E-mail address: jordi.genemola@udl.cat

Interest in vision techniques for Precision Agriculture has grown in recent years. Such techniques have contributed to triggering improvements in field conditions and have also helped farmers to better estimate their production through the use of fruit counting (Gené-Mola et al., 2020a) or fruit size estimation methods (Casagrande et al., 2021; Tsoulias et al., 2020), among others. This project focuses on the use of Deep Neural Networks (DNNs) for the detection of fruits on the tree and the estimation of their size.

Fruit growing and production are of great importance worldwide and in-field fruit monitoring contributes to the optimisation of its management (Anderson et al., 2021). Some of the most interesting tasks for farmers in this regard are fruit detection since it is used for fruit counting, and fruit size estimation, which is an important fruit quality parameter. In-field fruit counting and sizing also serve to estimate yield load and plan for its future transport, determine whether an automated harvesting system can be supported, or assess the validity of different cultivation techniques (Longchamps et al., 2022). Traditionally, in the agriculture industry, yield prediction of orchards has been a challenging task since fruit measurement is usually done manually using a Vernier calliper. Therefore, only an approximation is obtained, since not all the fruits are measured due to the time-consuming nature of the task. Another challenge is the generalization capacity of the methodology so fruits can be measured at different growth stages (Neupane et al., 2023).

Nowadays, most fruit detection works are based on Neural Networks, either using object detectors (Aguiar et al., 2021; Ghiani et al., 2021), semantic segmentation (Afonso et al., 2020; Peng et al., 2021) or instance segmentation (He et al., 2017; Liu et al., 2019). In contrast, most fruit size estimation methods are based on classical techniques such as geometrical (circle, ellipse, sphere…) fitting algorithms (Gene-Mola et al., 2023; Kurtser et al., 2020; Neupane et al., 2022) or segmenting the detected fruits and measuring the segmented area (Apolo-Apolo et al., 2020; Costa et al., 2021; Lu et al., 2022). These methods are highly affected by the quality of the fruit segmentation and the amount of fruit visibility (Wang et al., 2020), in addition, most of them are 3D-based, therefore are computationally expensive algorithms which limit real-time operation (Gené-Mola et al., 2021a). Methods based on 2D images require the usage of calibration targets that must be placed at the same distance to the cameras as the fruits, which adds complexity to the data acquisition process (Lu et al., 2022; Wang et al., 2018).

Alternatively, in this work, we propose a novel Multitask Neural Network specifically designed for simultaneous fruit detection and sizing using RGB-D images captured in-field conditions and without the need of requiring calibration targets. To the best of the authors' knowledge, this is the first end-to-end trainable Multitask DNN to detect and estimate the diameter of fruits by combining two architectures: one for detection and the other for diameter regression.

The paper itself is divided into the following sections. First, Section 2 describes the proposed methodology, presenting the dataset and how to pre-process it. An explanation is then provided of how and why depth maps are used, and finally, the developed Neural Network architecture is presented in detail. Section 3 presents and analyses the most relevant validation and test results in terms of fruit detection and size estimation. In Section 4, we discuss the results, while the main conclusions and future research lines are discussed in Section 5.

2. Materials and Methods

2.1. *Data and data pre-processing*

The data used in this project was generated by annotating some images from the PFuji-Size dataset (Gené-Mola et al., 2021b). The original dataset includes: (1) raw images used to generate the 3D point clouds of apple trees using structure from motion (SfM) and multi-view stereo (MVS) techniques; (2) the resulting 3D point clouds of the Fuji apple trees; (3) 3D instance (fruit) segmentation annotations; (4) fruit size (diameter) annotations which were manually obtained by measuring the maximum horizontal diameter using a Vernier calliper; and (5) the apples centre position. In addition, the fruit diameter and centre position were used to obtain 3D spherical mask of each apple. Part of the data was captured in October 2018, when apples were at an advanced ripening stage (growth stage of BBCH85 in Meier (2001) scale)(Fig. 1a), while the rest of the data was acquired in July 2020, when the apples were at 70% of their final size (growth stage of BBCH77 in Meier (2001) scale) (Fig. 1b). Each apple was assigned a unique identifier, which helped to manage the data.

(a) (b)

Fig. 1. (a) Sample image from data acquired in October 2018, when apples were red and at an advanced ripening stage (BBCH85). (b) Sample image from data acquired in July 2020, when apples were green and at 70% of their final size (BBCH77).

To adapt the original dataset to the needs of the present research, additional data curation was required to: (1) generate depth images; and (2) obtain 2D image annotations (apple masks and establish the correspondence between the mask and the ground truth diameter using the apple IDs). In order to generate depth images, SfM and MVS techniques were applied. To

this end, the same Metashape project (Agisoft Professional Metashape software, v1.6.4, St. Petersburg, Russia) used by Gené-Mola et al. (2021a) for 3D point cloud reconstruction was used to export the depth images registered with raw RGB images, obtaining the set of RGB-D images (Fig. 2).

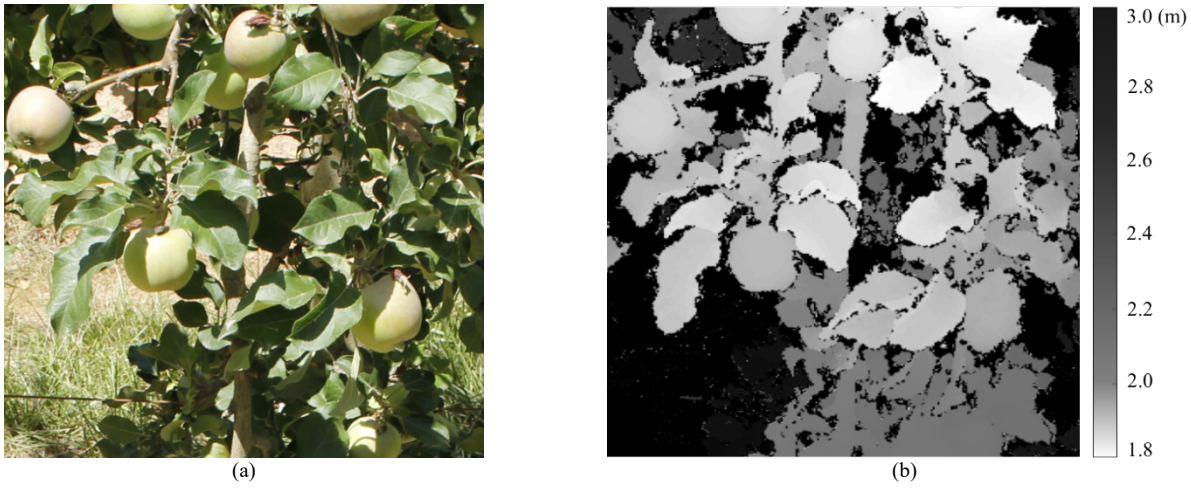
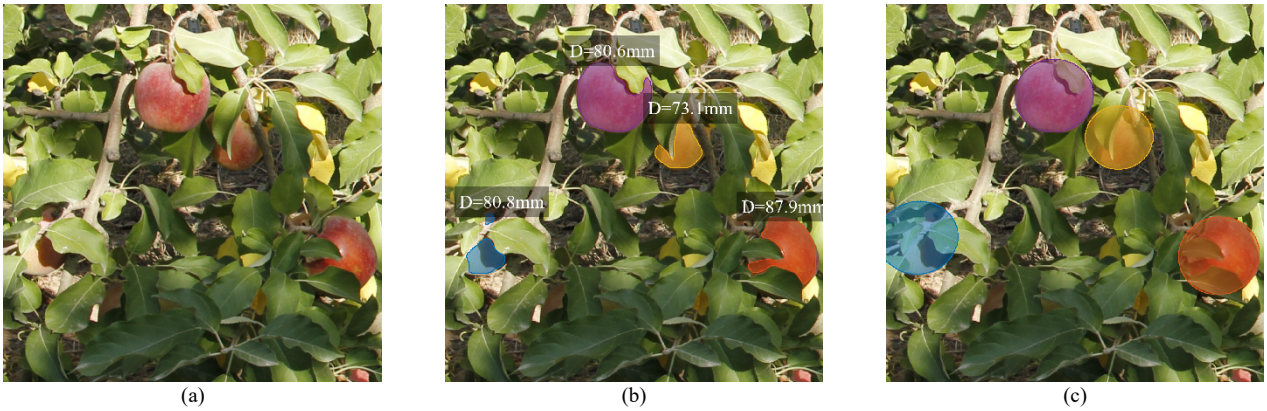


Fig. 2. RGB-D image from the 2020 set: (a) colour image; (b) depth image. The grey scale bar illustrates the depth values ranging from 1.8 m to 3.0 m.

Then, RGB-D images were annotated with: (1) instance segmentation masks (Fig. 3b); (2) apple diameter ground truth (associated with each mask) (Fig. 3b); and (3) 2D projection of each 3D spherical mask (Fig. 3c). To this end a semi-automatic annotation procedure was carried out. The apple IDs and the spherical masks were in the 3D space, but since this project is image-based (therefore, 2D), some adjustments were made. First, a Mask R-CNN (He et al., 2017) network trained with the Fuji-SfM dataset (Gené-Mola et al., 2020b) was used to automatically segment apples from the dataset images. Then, the 3D information was projected onto the 2D images in order to obtain the apple ID and consequently, its diameter provided in the PFuji-Size dataset. Furthermore, the 2D projection of the spherical masks was also obtained (Fig. 3c). This projection was carried out following the pinhole camera model (Faugeras, 1993). This allowed us to estimate the percentage of visibility V of each annotated apple, which is defined as the ratio between the area (in pixels) of the instance segmentation mask S_i (Fig. 3b) and the area of the projected spherical mask S_s (Fig. 3c). It can be translated to the following expression, which represents the visibility of an apple with a unique ID as the number of pixels of the instance mask (which represents the visible part of the apple) over the number of pixels in the projected spherical mask (which represents the whole area of the fruit):

$$V = \frac{S_i}{S_s} \times 100 \ [\%] \quad (1)$$

Finally, the result of this automatic annotation was manually corrected by using the VIA annotation software (Dutta and Zisserman, 2019). The manual correction consisted of: (1) deleting apple masks wrongly identified; (2) correcting the apple IDs and ground truth diameters wrongly matched; (3) labelling miss-annotated apples. The result of this annotation is shown in Fig. 3.



(a) (b) (c)

Fig. 3. Data annotation. (a) Sample image from data acquired in October 2018 (BBCH85). (b) Instance segmentation masks and apple diameter ground truth. (c) Projected spherical apple masks.

The generated dataset was split into training, validation and test sets. Images acquired from the west side of the row of trees were used for training, while east-side images were used for validation and testing. Table 1 details the number of images and apple annotations from BBCH77 (green apples)and BBCH85 (ripe apples) data split into the training, validation and test sets. All the data generated and used for this project (RGB-D images and annotations) has been made publicly available at http://www.grap.udl.cat/en/publications/papple_rgb-d-size-dataset/.

Table 1. Training, validation and test split: number of images and apple annotations from BBCH77 (green apples) and BBCH85 (ripe apples) data included in each split.

	Num. of images		Num. of apple annotations	
	BBCH77	**BBCH85**	**BBCH77**	**BBCH85**
Training set	1036	1268	3819	4760
Validation set	275	539	1161	2288
Test set	249	558	857	2450

2.2. *Multitask deep neural network*

2.2.1. *Instance segmentation branch*

Mask R-CNN (He et al., 2017) architecture was used as the baseline. It is a well-known two-stage network that detects objects in an image while also generating a segmentation mask for each detection. It is an extension of the Faster R-CNN (Ren et al., 2017) network and, in our case, we used the Detectron2 implementation (Wu et al., 2019) as the baseline. A representation of the Mask R-CNN architecture is shown in Fig. 4b (region coloured in orange). The main parts of this architecture are:

- Backbone (ResNet and Feature Pyramid Network): the Mask R-CNN implemented in this work uses the standard ResNet (He et al., 2016) architecture for encoding the image, where, at every layer, feature map size is reduced by half and the number of feature maps is doubled. Using ResNet-50 architecture, we extracted features from four feature maps ($res1$, $res3$, $res4$, $res5$). Next, to generate the final feature maps, the Feature Pyramid Network (FPN) was used. Identifying the same object at different scales is known to be a challenging task and sometimes the

network is not able to generalise well in this regard. For this reason, FPNs are of great use in such situations, since they take the input features at different scales and transform them so that, at smaller scales, the network focuses on the larger objects, while, at bigger scales, the network can focus on extracting features for the smaller objects. The outputs of the backbone are the final feature maps $fpn2$, $fpn3$, $fpn4$ and $fpn5$, while the feature map $fpn6$ is generated from a max-pooling operation on $fpn5$.

- A Region Proposal Network (RPN) is the core component of the R-CNN detector. The inputs are the feature maps coming from the FPN. Then, at every cell of the feature maps, it makes k predictions for k anchor boxes. It computes two things:
    - Regression to estimate the four coordinates of the proposed bounding boxes.
    - Binary classification: probability map of object existence in each cell.

  The k proposals are parameterised relatively to the k reference boxes (anchors) generated with cluster analysis over the training set. Mask R-CNN, as Faster R-CNN, uses three scales and three aspect ratios by default, yielding $k = 9$ anchors at each cell. The output boxes of the RPN are called proposal boxes.

- Box Head: The region of interest (RoI) Align process crops the rectangle regions of the feature maps that are specified by the proposal boxes. RoI Align is a more precise way to perform RoIPooling as it matches the feature map level that is most convenient for each bounding box. The result is fed to the Box Head, which has two fully-connected layers and performs regression to obtain the final corners of each bounding box. L1-loss is used to calculate the error.

- Mask Head: With the new bounding boxes estimated by the Box Head, RoI Align is again performed, and the output is the input for the Mask Head. This branch of the network is formed by three convolutional layers and a deconvolutional one. The loss used is the cross-entropy loss.

Usually, the input of Mask R-CNN is a 3-channel colour image. However, this project considers the depth information to be relevant for fruit size estimation purposes, so the input ends up having four channels: RGB+D (Fig. 4a). In order to boost speed, the depth channel was added following an early-fusion strategy (Sa et al., 2016). Due to this additional channel, filters from the first convolutional layer increase in depth (from 3 to 4). This modification does not affect the detection accuracy, as stated in previous works (Gené-Mola et al., 2019) and it is a way of simulating a tri-dimensional space using bi-dimensional data, which will be very helpful for the fruit size estimation.

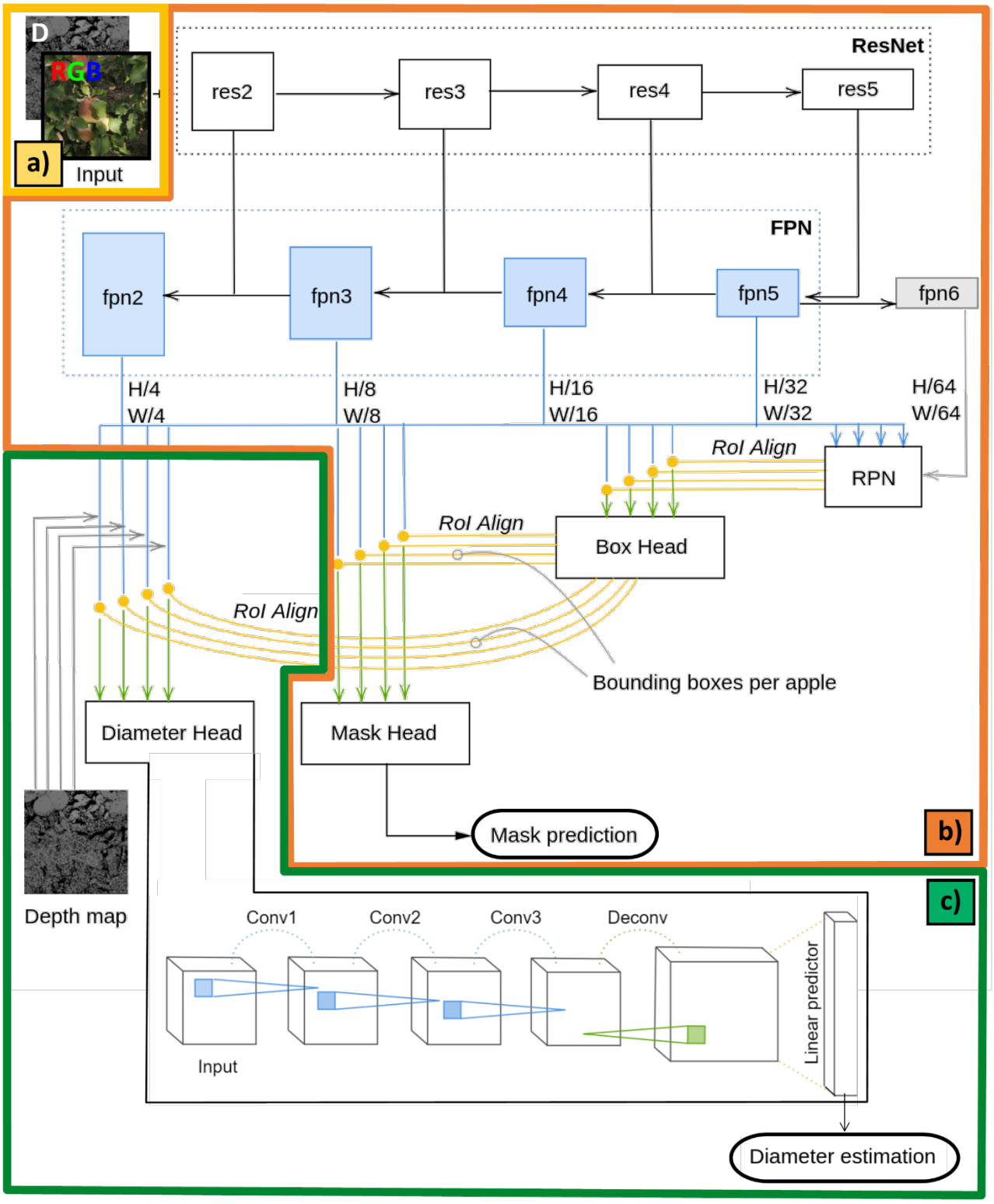


Fig. 4. Scheme of the Multitask Neural Network developed for fruit detection and size estimation. a) The network input allows up to 4 channel input images (RGB-D images). b) Instance segmentation architecture based on Mask R-CNN. c) Diameter head to estimate the size of each detected fruit by combining previous feature maps and the depth image using a set of convolutions and regression layers.

2.2.2. *Diameter regression branch*

The Diameter Head is a regression branch added to the baseline (Mask R-CNN) that aims to estimate the maximum horizontal diameter of detected apples. Fig. 4c (region coloured in green) illustrates a conceptual representation of the architecture of this branch inspired by the Mask Head. Its main differences from the Mask Head are the addition of depth information as an input and a final linear layer to predict the diameter.

The input of this Diameter Head comes from different parts of the network and has to be properly combined. The output from the FPN is fed into the sub-network. These four groups of feature maps (levels) have different sizes: $fp2(256 \text{ x } 256)$, $fp3$ (128 x 128),$fp4$ (64 x 64) and $fp5$ (32 x 32). In addition, to ensure the depth information influence plays a major role in the network's final weights, the depth maps of the corresponding images in the batch are concatenated to the FPN feature maps by reshaping them four times to the four desired sizes. The feature maps together with the resized depth information are matched with the bounding boxes coming from the Box Head using RoI Align.

The Diameter Head architecture (Fig. 4c) is formed by three 2D convolutions with a kernel of size 3 x 3, a stride of 1 x 1 and padding of 1 x 1. Each of the three convolution layers has a ReLU activation. Then, there is a deconvolution layer with a kernel of size 2 x 2 and a stride of 2 x 2 and a ReLU activation. This deconvolution, up-samples the image, which goes from 14 x 14 (default pooling resolution) to 28 x 28. After the deconvolution, the data is flattened and fed to a linear layer that predicts the diameter for that mask.

The developed network was implemented in the Pytorch framework and the code has been made publicly available jointly with the presented dataset at http://www.grap.udl.cat/en/publications/papple_rgb-d-size-dataset/.

2.2.3. *Network training and inference details*

a) *Weight initialisation*: Mask R-CNN has a set of weight initialisations pre-trained with different backbones on ImageNet(Deng et al., 2009). In our case, the used weights were pre-trained with a ResNet50 backbone. However, during the course of this project, we have carried out several additions to the baseline, and the new Diameter Head needs its set of pre-trained weights. We tried both a standard MSRA (He et al., 2015)initialization and re-using the Mask Head weights. Our experiments showed better results in re-using weights.

b) *Data augmentation*: One of the most popular techniques to increase the accuracy of the model is performing data augmentation. Creating "new" data from the existing images allows the models to generalise better and helps avoid overfitting. However, the scenario we are observing is quite monotonous, at least in the short term (and at certain hours of the day), so many augmentations such as 2D rotations might not be of great use. After some trial and error processes, we concluded that the best data augmentation technique was simply to apply a horizontal flip of the image.

c) *Evaluation metrics:* To evaluate the fruit detection results, we used precision (P), recall (R), F1-score, and average precision (AP) metrics. Diameter estimation was evaluated in terms of the mean absolute error (MAE), the mean bias error (MBE), the mean absolute percentage error (MAPE), the root mean square error (RMSE) and the coefficient of determination (R^2):

$$MAE = \frac{1}{m}\sum_{j=1}^{m}\left|D_{e_j} - D_{GT_j}\right|, \quad (1)$$

$$MBE = \frac{1}{m}\sum_{j=1}^{m}\left(D_{e_j} - D_{GT_j}\right), \quad (2)$$

$$MAPE = \frac{1}{m}\sum_{j=1}^{m}\frac{\left|D_{e_j} - D_{GT_j}\right|}{D_{GT_j}}, \quad (3)$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(D_{e_j} - D_{GT_j}\right)^2}, \quad (4)$$

$$R^2 = 1 - \frac{\sum_{j=1}^{m}\left(D_{e_j} - \overline{D_e}\right)^2}{\sum_{j=1}^{m}\left(D_{e_j} - f_j\right)^2}. \quad (5)$$

where m is the number of observations, D_e is the diameter estimation, D_{GT} is the diameter ground truth, $\overline{D_e}$ is the mean of estimated diameters and f is the linear regression model that relates D_e and D_{GT}.

d) *Training hyperparameter optimisation*: The Stochastic Gradient Descent (SGD) optimiser was used, and the optimal hyperparameters were found by means of a grid search. We considered a parameter to be optimal if the model yielded the smallest diameter error in the inference process. A grid search was performed over the following hyperparameters:

I. Learning rate: In this work, we performed a grid search over the following values: $lr = [2 \cdot 10^{-5},\ 2 \cdot 10^{-4},\ 2 \cdot 10^{-3}]$. The optimal lr was found to be $2 \cdot 10^{-4}$.

II. Batch size: In this work, we performed a grid search over the following values: $bs = [2,\ 4,\ 8]$. The optimal bs was 2.

III. RoIs per batch: This value represents the number of RoIs per training mini-batch. In this work, we performed a grid search over the following values: $roiBatch = [128,\ 512]$. The optimal $roiBatch$ was 128.

e) *Inference parameters:* The intersection over union (IoU) threshold used to determine if one detection is a true positive (TP) or false positive (FP) was set to$IoU = 0.5$, since it is the standard in the state-of-the-art. The other inference parameters were selected based on the validation set results. These parameters include:

I. Non-maximum suppression (NMS) threshold: NMS is used by the model to determine the number of accepted bounding box predictions since it will suppress boxes with an IoU bigger than the specified

threshold. We selected the optimum NMS threshold by finding the one that maximized the average precision (AP) in the validation set (Section 3.1). This analysis was carried out using the AP, as this metric is not affected by the selected confidence score.

II. Confidence score threshold: The minimum confidence score to consider a detection as positive was selected by analysing the P,R andF1-score curves for confidence values ranging from 0 to 1 (Section 3.1).

3. Results

3.1. Experiments on the validation set

The presented model was trained using the parameters detailed in the section above. Fig. 5a represents the loss curves for training and validation. The training was stopped when the model's generalisation capacity reached its limit, and the validation curve started getting flatter. The presented model does not show signs of overfitting as it can be observed in more detail in Fig. 5b and Fig. 5c, where both curves, for fruit detection and diameter estimation tasks, descend.

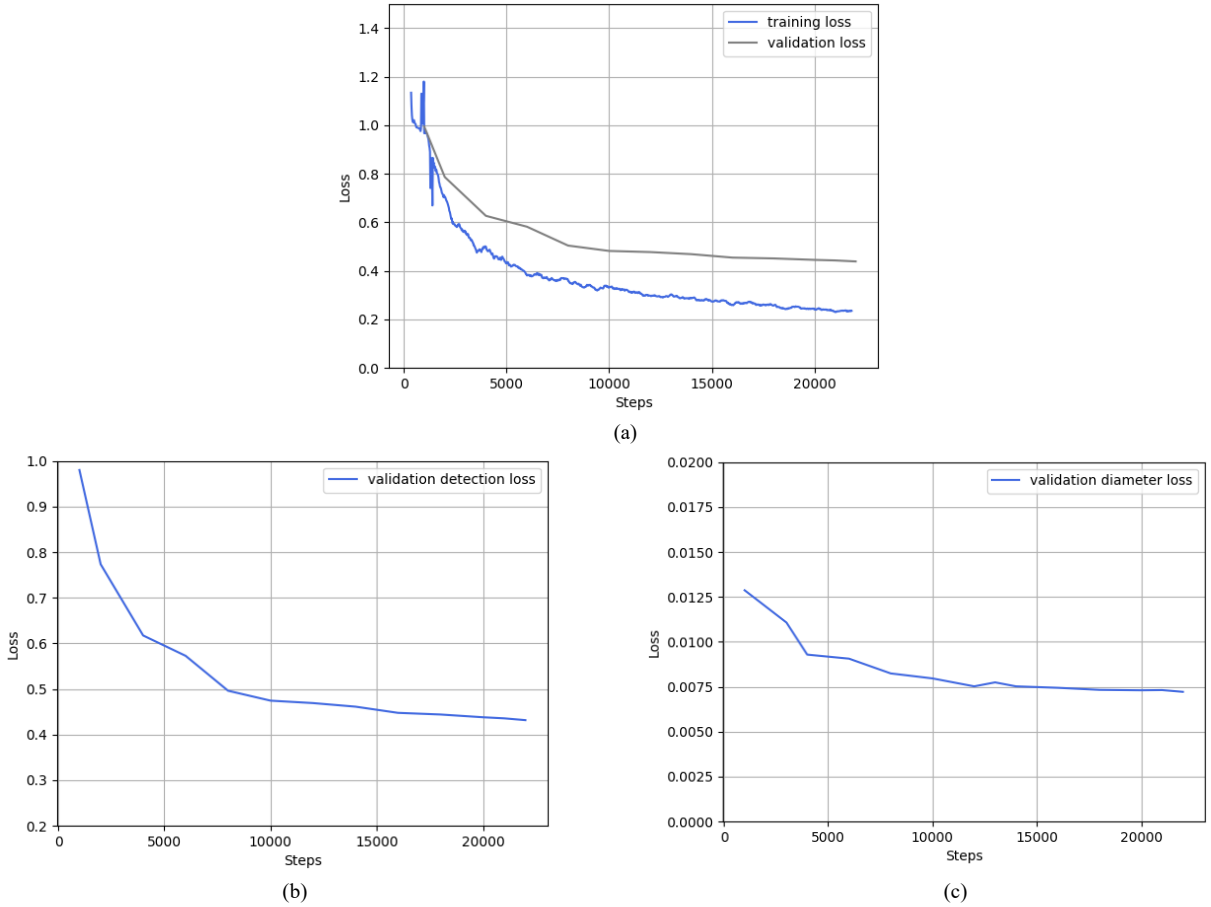
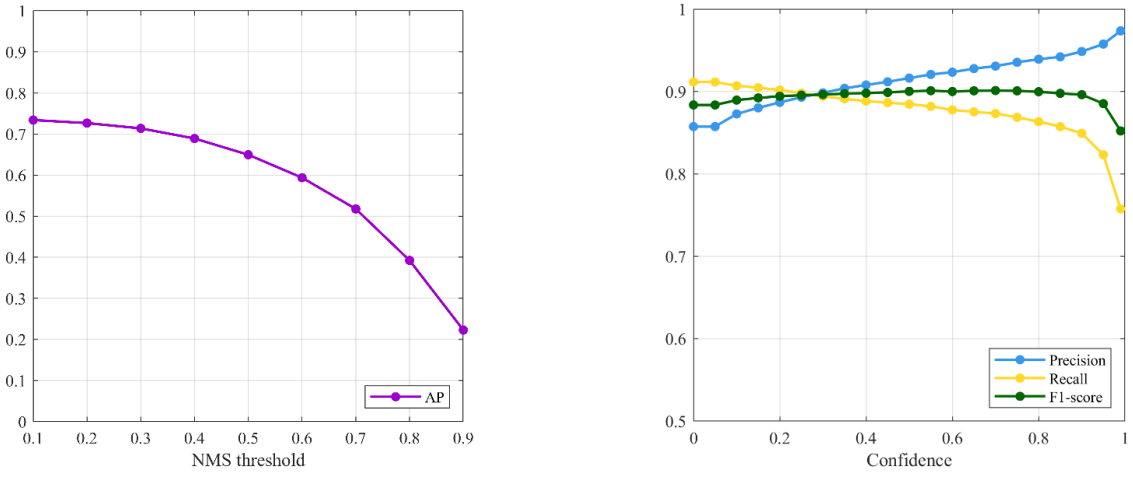


Fig. 5. From left to right and top to bottom: (a) Training and validation loss curves using the optimal parameters. (b) Validation loss curve for the fruit detection part of the network. (c) Loss curve for the diameter estimation task.

3.1.1. *Fruit detection*

To find the optimal NMS threshold, the AP metric was used to analyse the effect of applying different levels of NMS in the validation set. Fig. 6a represents the evolution of the AP curve regarding the NMS threshold. The highest values of AP correspond to the more restrictive NMS threshold ($NMS_{thresh} = 0.1$), which means that the bounding boxes that have more than 10% of overlap are eliminated. Based on these results, an NMS threshold of 0.1 was subsequently used to assess the fruit detection performance in the test set (Section 3.2).

The P, R and F1-score curves obtained in the validation set were analysed to choose the best confidence score value. Fig. 6b shows the behaviour of these parameters with respect to confidence. Note that the F1-score curve remains almost constant, although it slightly decreases with the confidence score. This might be due to the fact that the NMS threshold was already optimised, and so some possible outliers were already filtered. Furthermore, the P curve is an increasing function since the higher the confidence in the prediction, the less likely it is to encounter an FP. In contrast, the R curve tends to decrease, which is due to the fact that a more restrictive confidence threshold results in fewer predictions being accepted, and therefore, fewer TPs. Since the F1-score metric is maximised for a confidence value of 0.7, such confidence value was used to assess the fruit detection performance in the test set (Section 3.2).



(a) (b)

Fig. 6. Fruit detection results on the validation set. (a) average precision depending on the NMS threshold. (b) P, R and F1-score curves depending on the confidence score.

3.1.2. *Fruit size estimation*

The diameter estimation error is also affected by the degree of confidence in the prediction. The curve shown in Fig. 7a shows the evolution of the MAE of the estimated diameter with respect to the confidence threshold. The MAE decreases with higher confidence values, which means that diameter estimation improves when measuring fruits detected with higher confidence. This improvement is also observed with the increase of the coefficient of determination (R^2) between predicted and actual fruit sizes (Fig. 7b). Since the confidence value that minimizes the MAE is 0.99, the results presented in Section

3.2.2 were obtained using it as a threshold. Although 0.99 is restrictive, the number of fruits detected using it (about 2300 apples) is a representative sample of all detections (about 3150 apples).

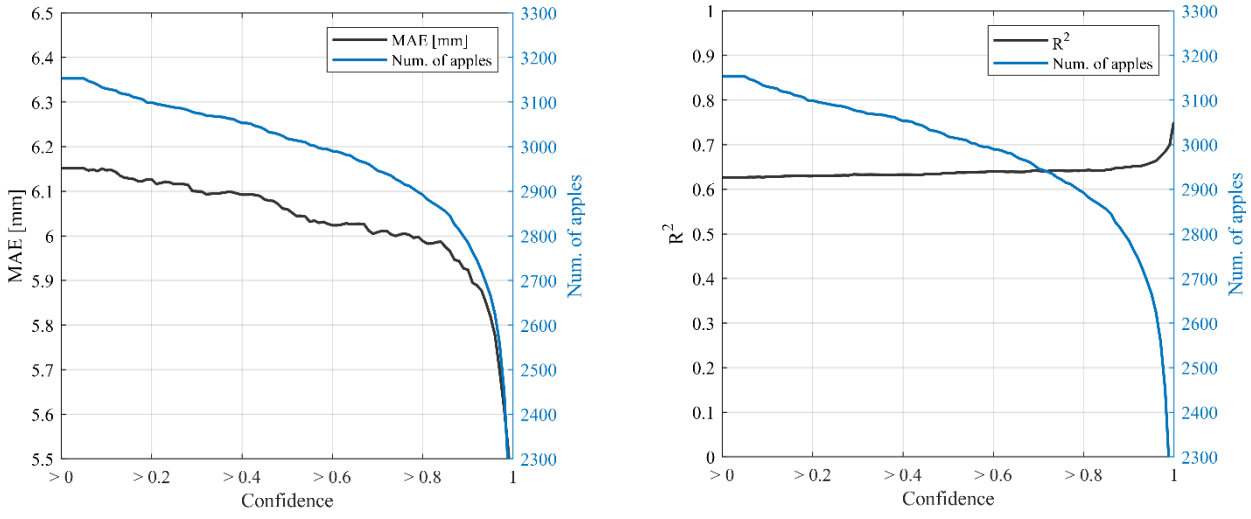


Fig. 7. Diameter estimation results on the validation set. (a) evolution of the MAE of diameter estimation and the number of considered apples at different confidence thresholds. (b) evolution of the coefficient of determination and the number of apples, also depending on the confidence threshold.

3.2. Test results

3.2.1. *Fruit detection*

The model was evaluated using the test set. Table 2 shows the detection results using the optimal parameters ($NMS_{thresh} = 0.1$, Confidence > 0.7). Similar F1-score results were obtained at different growth stages ($F1 = 0.88$). In terms of AP, the neural network presented a better performance detecting ripe apples ($AP_{BBCH85} = 0.75$) than detecting green apples ($AP_{BBCH77} = 0.69$). We attribute this difference to two main reasons: (1) the green colour of apples from the BBCH77 set makes the task more challenging due to the similarity of apple and leaf colour; and (2) the number of training samples in the BBCH85 set is larger than in the BBCH77 set.

Table 2. Fruit detection results in terms of Precision, Recall, F1-score and Average Precision (AP) depending on the growth stage where the images were taken and when evaluating the dataset.

	BBCH77	**BBCH85**	**BBCH77+BBCH85**
Precision	0.88	0.90	0.89
Recall	0.87	0.87	0.87
F1-score	0.88	0.88	0.88
AP	0.69	0.75	0.73

Fig. 8 shows the input image and the comparison between the actual ground truth and the model's prediction regarding both detection and diameter estimation. In terms of detection, the model performed as expected and the majority of fruits were detected. The critical cases were whenever there were high occlusions or the apples were in the margins of the image. In terms of computational speed, the average processing time per image was 0.1439 s/img, which corresponds to a throughput of 6.95 img/s. This processing times were obtained using an NVIDIA GeForce GTX 1080 Ti GPU.

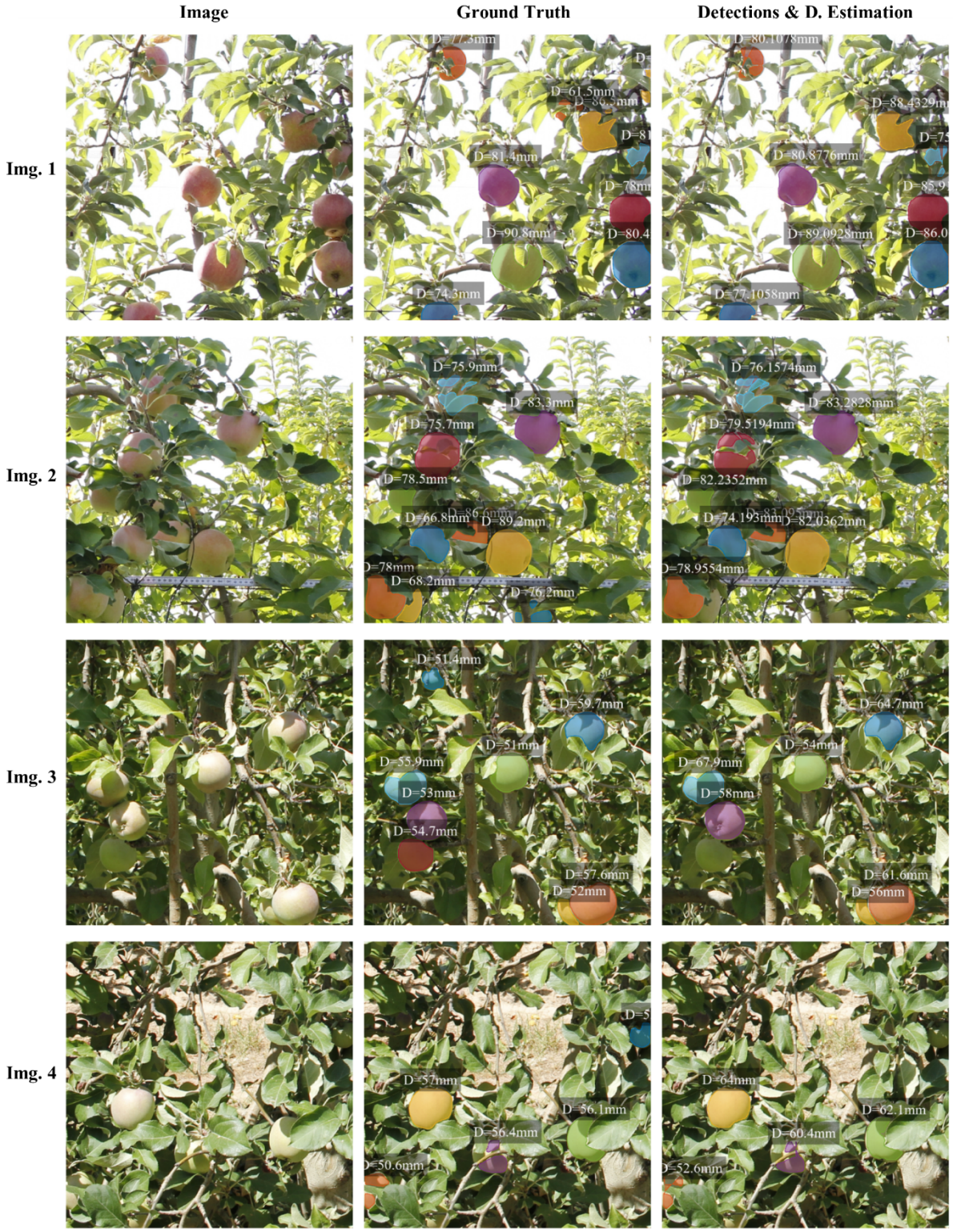


Fig. 8. Fruit detection and size estimation results in four randomly selected images. The top two rows show results on two images from the BBCH85 set (ripe apples), while the bottom two rows show results on images from the BBCH77 set (green unripe apples). The first column corresponds to the original images. The second column illustrates the instance segmentation masks and apple diameter ground truth. The third column illustrates the fruit detections (instance segmentation masks) and the estimated diameters.

3.2.2. *Fruit size estimation*

A detailed comparison of the performance of the model at different apple maturity stages can be found in Table 3. Results showed MAE between 4.16 mm and 6.22 mm, obtaining higher errors for ripe apples (with bigger sizes). The absolute error presented a standard deviation (σ) between 3.88 mm and 5.73 mm. This is considered a high standard deviation compared to the reported MAE. Authors attribute this high dispersion of the error to the higher errors obtained when measuring highly occluded apples. The percentage error was found to be similar at different growth stages (about 8%), concluding that the mean error is proportional to the size of the measured apples.

Table 3. MAE, σ, MBE, MAPE and RMSE of fruit diameter estimation according to the year the images were taken and when considering the whole dataset.

	BBCH77	**BBCH85**	**BBCH77+BBCH85**
MAE (mm)	4.16 mm	6.22 mm	5.64 mm
σ (mm)	3.88 mm	5.73 mm	5.35 mm
MBE (mm)	0.93 mm	-1.74 mm	0.98 mm
MAPE (%)	7.72 %	8.06 %	7.96 %
RMSE (mm)	5.69 mm	8.46 mm	7.77 mm

Fig. 9 compares the fruit size distribution obtained by measuring the fruits with the presented neural network with respect to the ground truth. The predicted distribution for both datasets is quite fitted to the actual ground truth. One thing to note is that, for the smaller apples, the tendency is to slightly overestimate their size (MBE $= 0.93$ mm). The opposite also happens with bigger apples (MBE $= -1.74$ mm). This effect is caused by the fact that the model was not trained for a specific apple size, so it tends to look for the middle ground.

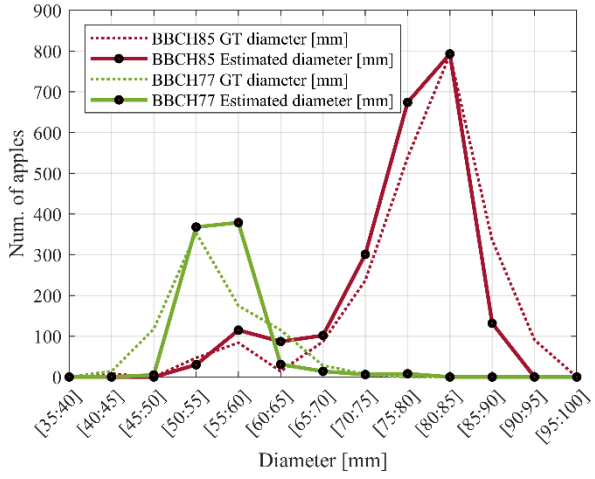


Fig. 9. Distribution of predicted and actual diameters in BBCH85(reddish ripe apples) and in BBCH77 (green unripe apples) sets.

Fig. 10 presents the effect of apple visibility on diameter estimation. As can be seen, the model behaves as expected, since the MAE gradually lowers when the visibility of the fruits increases. Optimal results were obtained when limiting the

measurement to apples with visibility greater than 65%. The linear correlation was also sensitive to occlusions, showing an increase in the coefficient of determination (R^2) when measuring highly visible apples. These results confirm that occlusion plays a significant role in the determination of the diameter distribution.

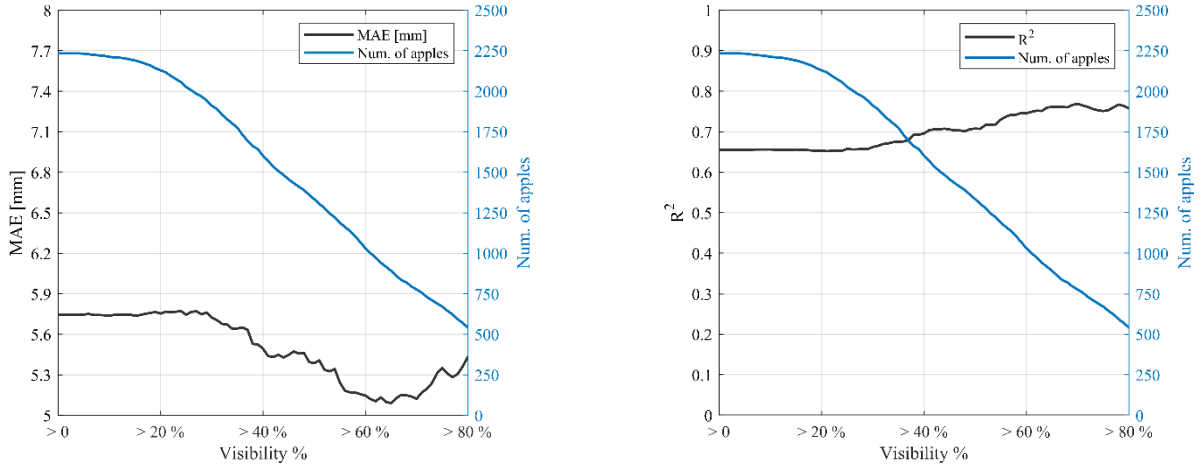


Fig. 10. Number of apples, MAE (left plot) and R² (right plot) depending on the level of visibility of the fruit.

4. Discussion

The presented model was robust enough to detect a significant number of apples at different growth stages and degrees of visibility, reporting an F1-score of 0.88 on the task. These results are comparable with other state-of-the-art works based on neural networks, which have reported F1-score values between 0.73 and 0.97 (Chu et al., 2021; Koirala et al., 2019; Wang and He, 2022). The apples that were not detected were highly occluded by other structural elements (leaves, trunks, other apples, …) or placed in the margins of the images with a small amount of the apple surface visible in the field of view of the camera. These fruit detection issues were also observed in previous works (Gené-Mola et al., 2019).

Having a robust fruit detector is of extreme importance for fruit counting but also for fruit sizing purposes, since it ensures that the size measures will be representative of the crop. The proposed methodology was able to predict the diameter of apples at different ripening stages, reporting a MAE of 5.64 mm. As presented in the previous section, it tends to overestimate the size of the smaller apples, and underestimate the size of the bigger ones, however, we argue that this bias is negligible since in Fig. 9 we showed that the diameter prediction distribution is adapting properly to the ground truth diameter distribution. The results we show might differ if images with very different lighting conditions or different tree-camera distances are used. Nevertheless, the data-gathering settings are easy to reproduce and are public alongside the whole dataset (Gené-Mola et al. (2021b)). Although it is difficult to compare methodologies tested with different datasets, we can state that, in terms of mean

diameter errors, our method performed similarly to other state-of-the-art methods, which reported MAE results between 3.5 mm and 12.4 mm, as we can see in Table 4.

Table 4.. Comparison of the proposed system with other state-of-the-art methods. Results are reported in terms of MAE and RMSE.

Authors	Methodology	MAE	RMSE
(Tsoulias et al., 2020)	LiDAR + max. distance	3.5 – 12.4 mm	N.A.
(Gené-Mola et al., 2021[a])	SfM + MVS + shape fitting	3.7 – 7.7 mm	5.1 – 12.5 mm
(Mengoli et al., 2022)	RGB+D + shape fitting	N.A.	7.9 – 8.6 mm
Proposed method	RGB+D + multitask DNN	4.16 – 5.64 mm	5.69 – 7.77 mm

N/A = not available.

The main contribution of this work is that, for the first time, an end-to-end deep learning architecture has been designed and tested for the simultaneous detection and measurement of fruits. Besides its performance in terms of detection and sizing, the method presents other significant advantages: it overcomes the limitations of traditional sizing methodologies where calibration targets are required and must be placed at the same distances from the fruits to be measured (Lu et al., 2022; Wang et al., 2018). With these techniques, only the fruits around the calibration targets can be measured, while our proposed methodology could measure larger areas efficiently. In addition, previous fruit sizing works required the identification of feature points on the apple images to subsequently perform a geometrical measurement (Wang et al., 2020). Alternatively, the presented method directly estimates the diameter of the apples without the need to identify specific key points to measure, which results in a more efficient method. Furthermore, since it is based on a CNN that can be processed with graphic processing units (GPUs) and parallel computing, our method will permit its use for real-time and edge-computing applications (Mazzia et al., 2020). We obtained a competitive throughput of 6.95 $\mathrm{img/s}$ thanks to using an early-fusion network architecture. This is considered a high inference speed for simultaneous fruit detection and sizing compared with other state-of-the-art methods. For instance, Luo et al. (2021) required 17.88s for Red grape detection. Rojas-Cid et al. (2019) designed a system able to measure 16 mangoes per minute, while Tsoulias et al. (2020) developed an apple detection and sizing method based on LiDAR that required a processing time of 13s per tree. Another advantage of our method is that it is based on the use of RGB-D images, which allows us to include the 3D information without the computational complexity of adding another dimension. In this paper, we used highly precise depth maps that were created using SfM which requires a great number of images of the studied area. We propose to study the different effects that the depth maps obtained using commercial sensors have on the presented framework in future work.

Some fruit sizing works from the literature limit the evaluation of their methods on fully visible fruits (Gongal et al., 2018; Herrero-Huerta et al., 2015; Wang et al., 2020, 2017). However, the present work presents an analysis of results at different fruit visibility percentages. Results showed that the more visible an apple is, the better its diameter will be predicted. The

MAE improved from 5.64 mm to 5.09 mm when limiting the measurement to fruits with visibility percentages higher than 65%. This suggests that future works should explore the development of a method for automatically identifying the most visible apples and not consider the prediction of low visibility scores.

5. Conclusions

This project proposes a deep learning approach for simultaneous fruit detection and size estimation. The method presented can be used to measure fruits at different growth stages and, as stated in the introduction, such insights can provide farmers with much-needed data to manage their crops more efficiently. The baseline for this work was the Mask R-CNN instance segmentation network, which was extended with a regression branch in order to compute the diameter of the detected apples, yielding successful results both in terms of fruit detection and fruit size estimation.

Regarding apple detection, our method achieves state-of-the art performance, with an F1-score of 0.88. Furthermore, the presented architecture was able to estimate fruit size with a MAE of 5.64 mm. Results were robust at different degrees of visibility but, when discarding the measured highly occluded apples, the correlation between actual and estimated diameter slightly improved (from $R^2 = 0.66$ to $R^2 = 0.77$). These results are similar to other state-of-the-art methodologies, but our proposed method has the following advantages: a) it simultaneously detects and estimates the size with a single end-to-end trainable network; b) it is efficient and fast so it can be used for real-time applications, and c) it uses RGB-D data which can be acquired with affordable depth cameras.

The method presented successful results, demonstrating the promising future of deep learning approaches in the field of fruit sizing. However, there is still room for improvement. A combination of the proposed method with automatic estimation of fruit visibility would help to select the best candidate apples to be measured. In addition, an unexplored and promising path for fruit size computation would be to use Graph Neural Networks, which use 3D data. Finally, although this work deals with apples, it could be extended to other fruit varieties.

Acknowledgements

This work was partly funded by the Departamentde Recercai Universitats de la Generalitat de Catalunya (grant 2021 LLAV 00088), the Spanish Ministry of Science, Innovation and Universities (grants RTI2018-094222-B-I00[PAgFRUIT project], PID2021-126648OB-I00 [PAgPROTECT project] and PID2020-117142GB-I00 [DeeLight project] by MCIN/AEI/10.13039/501100011033 and by “ERDF, a way of making Europe”, by the European Union).The work of Jordi

Gené Mola was supported by the Spanish Ministry of Universities through a Margarita Salas postdoctoral grant funded by the European Union - NextGenerationEU.

REFERENCES

Afonso, M., Fonteijn, H., Fiorentin, F.S., Lensink, D., Mooij, M., Faber, N., Polder, G., Wehrens, R., 2020. Tomato Fruit Detection and Counting in Greenhouses Using Deep Learning. Front. Plant Sci. 11, 1759. https://doi.org/10.3389/FPLS.2020.571299/BIBTEX

Aguiar, A.S., Magalhães, S.A., Dos Santos, F.N., Castro, L., Pinho, T., Valente, J., Martins, R., Boaventura-Cunha, J., 2021. Grape Bunch Detection at Different Growth Stages Using Deep Learning Quantized Models. Agron. 2021, Vol. 11, Page 1890 11, 1890. https://doi.org/10.3390/AGRONOMY11091890

Anderson, N.T., Walsh, K.B., Wulfsohn, D., 2021. Technologies for forecasting tree fruit load and harvest timing—from ground, sky and time. Agronomy 11. https://doi.org/10.3390/agronomy11071409

Apolo-Apolo, O.E., Martínez-Guanter, J., Egea, G., Raja, P., Pérez-Ruiz, M., 2020. Deep learning techniques for estimation of the yield and size of citrus fruits using a UAV. Eur. J. Agron. 115, 126030. https://doi.org/10.1016/j.eja.2020.126030

Casagrande, E., Génard, M., Lurol, S., Charles, F., Plénet, D., Lescourret, F., 2021. A process-based model of nectarine quality development during pre- and post-harvest. Postharvest Biol. Technol. 175. https://doi.org/10.1016/j.postharvbio.2020.111458

Chu, P., Li, Z., Lammers, K., Lu, R., Liu, X., 2021. Deep learning-based apple detection using a suppression mask R-CNN. Pattern Recognit. Lett. 147, 206–211. https://doi.org/10.1016/j.patrec.2021.04.022

Costa, L., Ampatzidis, Y., Rohla, C., Maness, N., Cheary, B., Zhang, L., 2021. Measuring pecan nut growth utilizing machine vision and deep learning for the better understanding of the fruit growth curve. Comput. Electron. Agric. 181, 105964. https://doi.org/10.1016/J.COMPAG.2020.105964

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. https://doi.org/10.1109/CVPRW.2009.5206848

Dutta, A., Zisserman, A., 2019. The VIA Annotation Software for Images, Audio and Video, in: Proceedings of the 27th ACM International Conference on Multimedia. ACM, New York, NY, USA. https://doi.org/10.1145/3343031.3350535

Faugeras, O., 1993. Three-dimensional computer vision : a geometric viewpoint, Cambridge, MA, USA. MIT Press.

Gene-Mola, J., Ferrer-Ferrer, M., Gregorio, E., Blok, P.M., Hemming, J., Morros, J.-R., Rosell-Polo, J.R., Vilaplana, V., Ruiz-Hidalgo, J., 2023. Looking behind occlusions: a study on amodal segmentation for robust on-tree apple fruit size estimation. Comput. Electron. Agric. In Press.

Gené-Mola, J., Sanz-Cortiella, R., Rosell-Polo, J.R., Escolà, A., Gregorio, E., 2021a. In-field apple size estimation using photogrammetry-derived 3D point clouds: comparison of 4 different methods considering fruit occlusions. Comput. Electron. Agric. 188, 106343. https://doi.org/https://doi.org/10.1016/j.compag.2021.106343

Gené-Mola, J., Sanz-Cortiella, R., Rosell-Polo, J.R., Escolà, A., Gregorio, E., 2021b. PFuji-Size dataset: A collection of images and photogrammetry-derived 3D point clouds with ground truth annotations for Fuji apple detection and size estimation in field conditions. Data Br. 39. https://doi.org/10.1016/j.dib.2021.107629

Gené-Mola, J., Sanz-Cortiella, R., Rosell-Polo, J.R., Morros, J.-R.R., Ruiz-Hidalgo, J., Vilaplana, V., Gregorio, E., 2020a. Fruit detection and 3D location using instance segmentation neural networks and structure-from-motion photogrammetry. Comput. Electron. Agric. 169. https://doi.org/https://doi.org/10.1016/j.compag.2019.105165

Gené-Mola, J., Sanz-Cortiella, R., Rosell-Polo, J.R., Morros, J.R., Ruiz-Hidalgo, J., Vilaplana, V., Gregorio, E., 2020b. Fuji-SfM dataset: A collection of annotated images and point clouds for Fuji apple detection and location using structure-from-motion photogrammetry. Data Br. 30. https://doi.org/10.1016/j.dib.2020.105591

Gené-Mola, J., Vilaplana, V., Rosell-Polo, J.R., Morros, J.-R.R., Ruiz-Hidalgo, J., Gregorio, E., 2019. Multi-modal deep learning for Fuji apple detection using RGB-D cameras and their radiometric capabilities. Comput. Electron. Agric. 162, 689–698. https://doi.org/10.1016/j.compag.2019.05.016

Ghiani, L., Sassu, A., Palumbo, F., Mercenaro, L., Gambella, F., 2021. In-Field Automatic Detection of Grape Bunches under a Totally Uncontrolled Environment. Sensors 2021, Vol. 21, Page 3908 21, 3908. https://doi.org/10.3390/S21113908

Gongal, A., Karkee, M., Amatya, S., 2018. Apple fruit size estimation using a 3D machine vision system. Inf. Process. Agric. 5, 498–503. https://doi.org/10.1016/j.inpa.2018.06.002

He, K., Gkioxari, G., Dollar, P., Girshick, R., 2017. Mask R-CNN. Proc. IEEE Int. Conf. Comput. Vis. 2017-Octob, 2980–2988. https://doi.org/10.1109/ICCV.2017.322

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 770–778. https://doi.org/10.1109/CVPR.2016.90

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proc. IEEE Int. Conf. Comput. Vis. 2015 Inter, 1026–1034. https://doi.org/10.1109/ICCV.2015.123

Herrero-Huerta, M., González-Aguilera, D., Rodriguez-Gonzalvez, P., Hernández-López, D., 2015. Vineyard yield estimation by automatic 3D bunch modelling in field conditions. Comput. Electron. Agric. https://doi.org/10.1016/j.compag.2014.10.003

Koirala, A., Walsh, K.B., Wang, Z., McCarthy, C., 2019. Deep learning – Method overview and review of use for fruit detection and yield estimation. Comput. Electron. Agric. 162, 219–234. https://doi.org/10.1016/j.compag.2019.04.017

Kurtser, P., Ringdahl, O., Rotstein, N., Andreasson, H., 2020. PointNet and geometric reasoning for detection of grape vines from single frame RGB-D data in outdoor conditions. Proc. North. Light. Deep Learn. Work. 1, 6. https://doi.org/10.7557/18.5155

Liu, X., Zhao, D., Jia, W., Ji, W., Ruan, C., Sun, Y., 2019. Cucumber fruits detection in greenhouses based on instance segmentation. IEEE Access 7, 139635–139642. https://doi.org/10.1109/ACCESS.2019.2942144

Longchamps, L., Tisseyre, B., Taylor, J., Sagoo, L., Momin, A., Fountas, S., Manfrini, L., Ampatzidis, Y., Schueller, J.K., Khosla, R., 2022. Yield sensing technologies for perennial and annual horticultural crops: a review. Precis. Agric. 23, 2407–2448. https://doi.org/10.1007/s11119-022-09906-2

Lu, S., Chen, W., Zhang, X., Karkee, M., 2022. Canopy-attention-YOLOv4-based immature/mature apple fruit detection on dense-foliage tree architectures for early crop load estimation. Comput. Electron. Agric. 193, 106696. https://doi.org/10.1016/J.COMPAG.2022.106696

Luo, L., Liu, W., Lu, Q., Wang, J., Wen, W., Yan, D., Tang, Y., 2021. Grape berry detection and size measurement based on edge image processing and geometric morphology. Machines 9. https://doi.org/10.3390/machines9100233

Mazzia, V., Khaliq, A., Salvetti, F., Chiaberge, M., 2020. Real-time apple detection system using embedded systems with hardware accelerators: An edge AI application. IEEE Access 8, 9102–9114. https://doi.org/10.1109/ACCESS.2020.2964608

Meier, U., 2001. Growth stages of mono- and dicotyledonous plants, BBCH Monograph. https://doi.org/10.5073/bbch0515

Mengoli, D., Bortolotti, G., Piani, M., Manfrini, L., 2022. On-line real-time fruit size estimation using a depth-camera sensor. 2022 IEEE Work. Metrol. Agric. For. MetroAgriFor 2022 - Proc. 86–90. https://doi.org/10.1109/MetroAgriFor55389.2022.9964960

Neupane, C., Koirala, A., Walsh, K.B., 2022. In-Orchard Sizing of Mango Fruit: 1. Comparison of Machine Vision Based Methods for On-The-Go Estimation. Horticulturae 8. https://doi.org/10.3390/horticulturae8121223

Neupane, C., Pereira, M., Koirala, A., Walsh, K.B., 2023. Fruit Sizing in Orchard : A Review from Caliper to Machine Vision with Deep Learning. Sensors (Basel). 23, 3868. https://doi.org/https://doi.org/10.3390/s23083868

Peng, Y., Wang, A., Liu, J., Faheem, M., 2021. A Comparative Study of Semantic Segmentation Models for Identification of Grape with Different Varieties. Agric. 2021, Vol. 11, Page 997 11, 997. https://doi.org/10.3390/AGRICULTURE11100997

Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39, 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Rojas-Cid, J.D., Perez-Bailon, W., Rosas-Arias, L., Roman-Ocampo, D.B., Lopez-Tello, J.A., 2019. Design of a size sorting machine based on machine vision for mexican exportation mangoes, in: 2018 IEEE International Autumn Meeting on Power, Electronics and Computing, ROPEC 2018. https://doi.org/10.1109/ROPEC.2018.8661378

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

Tsoulias, N., Paraforos, D.S., Xanthopoulos, G., Zude-Sasse, M., 2020. Apple shape detection based on geometric and radiometric features using a LiDAR laser scanner. Remote Sens. 12, 2481. https://doi.org/10.3390/RS12152481

Wang, D., He, D., 2022. Fusion of Mask RCNN and attention mechanism for instance segmentation of apples under complex background. Comput. Electron. Agric. 196, 106864. https://doi.org/10.1016/j.compag.2022.106864

Wang, D., Li, C., Song, H., Xiong, H., Liu, C., He, D., 2020. Deep learning approach for apple edge detection to remotely monitor apple growth in orchards. IEEE Access 8, 26911–26925. https://doi.org/10.1109/ACCESS.2020.2971524

Wang, Z., Koirala, A., Walsh, K., Anderson, N., Verma, B., 2018. In field fruit sizing using a smart phone application. Sensors (Switzerland) 18. https://doi.org/10.3390/S18103331

Wang, Z., Walsh, K.B., Verma, B.B., 2017. On-tree mango fruit size estimation using RGB-D images. Sensors 17, 2738. https://doi.org/10.3390/s17122738

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., Girshick, R., 2019. Detectron2 [WWW Document]. GitHub Repos. URL https://github.com/facebookresearch/detectron2 (accessed 2.10.22).