Multimedia Tools and Applications (2023) 82:21657–21671
https://doi.org/10.1007/s11042-023-14668-8

Mask R-CNN for quality control of table olives

Miguel Macías-Macías[1] · Héctor Sánchez-Santamaria[2] · Carlos J. García Orellana[1] · Horacio M. González-Velasco[1] · Ramón Gallardo-Caballero[1] · Antonio García-Manso[1]

Received: 2 December 2021 / Revised: 9 June 2022 / Accepted: 3 February 2023 /
Published online: 18 February 2023

Abstract

In this paper we propose an object detector based on deep learning for scanning samples of table olives. For the construction of the system we have used a Mask R-CNN neural network. This network is able to segment the image providing a mask for each of the olives in the sample from which we can obtain the calibre of the object. In addition, the system is able to measure the degree of ripeness of the olives classifying them as green, semi-ripe and ripe, and identifying those fruits that are defective due to disease or damage caused by the harvesting process. The proposed system achieves success rates of 99.8% in the detection of olive fruits in photograms, 93.5% in the classification of fruit by ripeness and close to 80% in the detection of defects.

Keywords Object detection · Mask-RCNN · Deep-learning · Table olives

✉ Miguel Macías-Macías
mmacias@unex.es

Héctor Sánchez-Santamaria
sasah@unex.es

Carlos J. García Orellana
cjgarcia@unex.es

Horacio M. González-Velasco
hmgvelas@unex.es

Ramón Gallardo-Caballero
rgallardo@unex.es

Antonio García-Manso
agmanso@unex.es

[1] Instituto de Computación Científica Avanzada (ICCAEx), Universidad de Extremadura, E-06006 Badajoz, Spain

[2] Centro Universitario de Mérida (CUMe), Universidad de Extremadura, E-06800 Mérida, Spain

1 Introduction

Spain is the world's leading producer and exporter of table olives. According to the latest data published by the Food Information and Control Agency (AICA), as of January 31, 2021, the production of table olives for the 2020/2021 campaign amounted to 545,310 tons. The most cultivated varieties are: *hojiblanca, manzanilla* and *gordal* (Fig. 1).

The Trade Standard Applying to Table Olives (COI/OT/NC no. 12004) defines table olives as the product "*prepared from the sound fruits of varieties of the cultivated olive tree (Oleae europaea L.) that are chosen for their production of olives whose volume, shape, flesh-to-stone ratio, fine flesh, taste, firmness and ease of detachment from the stone make them particularly suitable for processing*".

Table olives are classified into one of the following types according to the degree of ripeness of the fresh fruits:

- Green olives: Fruits harvested during the ripening period, prior to colouring and when they have reached normal size. Once processed, the green olive colour may vary from green to straw-yellow.
- Semi-ripe olives: Fruits harvested before the stage of complete ripeness is attained, at colour change. After processing, this type of olive may vary from pink to rosé wine or brown [24].
- Ripe olives: Fruits harvested when fully ripe or slightly before full ripeness is reached. Once processed, ripe olives may range from reddish black to violet-black, deep violet, greenish black, or deep chestnut.

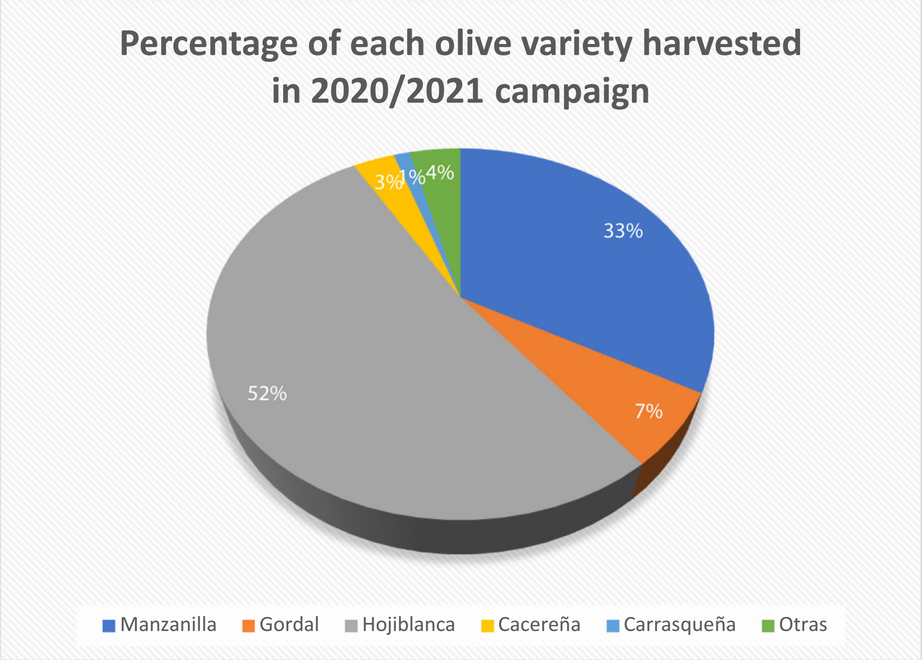

Fig. 1 The most cultivated varieties of table olives in Spain in the 2020/2021 campaign

Traditionally, to avoid damaging the fruit, olives intended for direct consumption are mostly hand-harvested from the olive tree using the "milking" technique. When they reach their maximum size and proper degree of ripeness –e.g., green to yellow-, between the months of September and November, olives, are deposited one by one in a basket that the collector carries. The "milking" could be complemented with machinery such as shakers, a kind of mechanical arm that facilitates the fall of the fruit onto sheets of cloth, placed below and around the olive tree so that the olives cannot come into contact with the ground. However, because mechanical harvesting causes a high percentage of damage to the fruit, manual harvesting is still widely used.

After harvesting, olives are immediately transported to the manufacturer to determine their quality and pricing. As measuring the whole harvested batch is impossible, "escandallo" is carried out from the study of samples and consists of two stages:

- Maturity index determination and defect identification. The quality of an olive fruit can be determined by its external appearance. The skin and flesh color allows us to specify the maturity index of the olives. As the olives mature, the colour of the outer part (*epicarp*) changes from bright green to purple-green, purple and finally black. The inner part (*pericarp*) changes from white-yellow to purple-black. Fruit defects (bruising, wrinkles, olives bitten by fruit flies, hail-damages, olives affected by cochineal insects) can also be visually detected. This defects decrease the final pricing of the product.
- Sizing. Olives are size-graded according to the number of fruits per kilogram or hectogram. The size scale, in one kilogramme, goes from 60/70, 71/80, 81/90, successively to 401/420 (called "perdigon" and used to produce olive oil).

Although the traditional "escandallo" (see Fig. 2), inspires confidence in olive growers, it is a slow and tedious process. Sometimes, it must be carried out several hours after taking the sample with the consequent deterioration of the olives and detriment to the olive growers' own interests. Therefore, there are manufacturing companies that are trying to automate the procedure with powerful equipment capable of managing a greater number of kilograms, which would also allow the process to go more quickly.

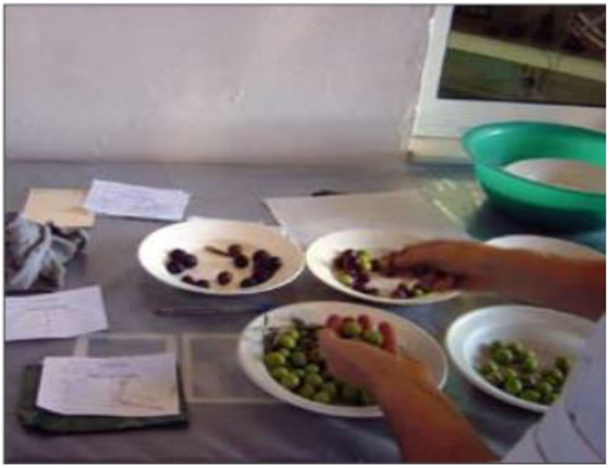

Fig. 2 Traditional "escandallo" of olive fruits

Other studies have referenced the automation of quality control in fruits and vegetables by applying mechanized, vision-based methods even in spectral ranges beyond the sensibility of the human eye, e.g., ultraviolet and near-infrared regions [4]. Recently, Piedad et al. [20] use RGB coordinates with machine learning to classify banana tiers in different commercial qualities. Kaur et al. [14] estimate the maturity of plumps using the RGB coordinates. Ramos et al. [22] present a system which automatically determines the ripeness percentage of the fruit on a coffee branch and its ripeness index through analysis of 3D information obtained with a monocular camera in outdoor environments and under uncontrolled lighting, contrast, and occlusion conditions. Munera et al. [19] analyze the quality of 'Mollar de Elche' intact pomegranate fruit and arils in a non-supervised way by means of PCA using colour and hyperspectral images.

Particularly with olive fruits, Ponce et al. [21] propose a system to estimate mass and size of "picual" and "arbequina" olive varieties, mainly dedicated to the production of olive oil. The system uses mathematical morphology and statistical thresholding to segment the acquired images. These images show olives spatially distributed on a white plastic mat. Guzmán et al. [10] present a system to estimate the maturity index of "picual" variety olives destined to produce olive oil. To do this, they use CIELAB coordinates of the olives. As the external appearance of an olive's skin is the most decisive factor in determining its quality, Riquelme et al. [26] describe a procedure to classify olives in eight categories according to external damage using three different discriminant analysis (DA). They only focus on the problem of image classification, not on the detection of olives. Aquino et al. [3] present a system for early yield estimation of olive orchards. It is based in the use of convolutional neural networks (CNN) capable of identifying olive fruits visible in images covering entire olive trees.

In this paper we present an automatic system for quality control of table olives based on deep learning. The developed system automatically detects, segments, and classifies table olives from an image taken of the product. It determines the average size of the olives in the sample and the percentage of defective fruits depending on their state of maturity or the damage caused to the skin. The system is low cost. It can run on a Raspberry Pi 4 Model B 4GB and is accessed and controlled remotely from a browser through Node-RED. The system can be installed in the product reception line and sample a wider set of olive fruits, improving the reliability of the quality control procedure. In addition, the system can be used for the automatic selection and elimination of defective olives, a process that is currently carried out manually.

We propose a system based on deep learning because in our previous experience in the field of pollen grain detection (also with ellipsoidal shape) the increase in performance of deep learning based techniques with respect to other classical segmentation techniques was very high. For example, comparing localization performance with the classical circular Hough transform we found that sensitivity improved by 18.9% and accuracy by 15.9%, for a total of 1235 objects in 135 images [7].

The remainder of this paper is organized as follows: Section 2 explains the materials and methods. Section 3 presents the results of the developed system, and Section 4 concludes this paper.

2 Material and methods

2.1 Prototypes set

Olive samples of the "manzanilla sevillana" variety were collected in the southwest province of Badajoz (Spain) in September 2020. These samples were deposited on a white tray and photographed with a Raspberry Pi Camera Module v2.1 connected to a Raspberry Pi 4 B. The distance between the tray and the camera was 50 cm. The images, captured with natural light, had a resolution of 1280 × 960 pixels.

From these images and to train our object detector, ground truth images were generated using VGG Image Annotator (VIA) software (https://www.robots.ox.ac.uk/~vgg/software/via/). VIA is a simple and standalone manual annotation software for image, audio and video that runs in a web browser. This software makes it possible to select a mask for each individual object detected in an image and associate it with different categories (attribute name), each of them with a set of classes (id name). For example, we can create the attribute "object" with the ID "olive", the attribute "ripeness" with the IDs "green", "semi-ripe" and "ripe" or the attribute "quality" with the IDs "damaged", "green", "semi-ripe" and "ripe". Figure 3 shows the masks selected with VIA for an image with 161 olive fruits together with their labelling corresponding to the "ripeness" attribute.

To adjust our models, we labelled a total of 50 images, each one with between 80 and 180 olive fruits, using the software mentioned above. These images were split into a training set of 40 images and a test set of 10.

2.2 The object detector model

Traditionally, an object detector consisted of an image classifier that was applied on small regions of the input image using the sliding window method [5]. A sliding window is a rectangular region of fixed width and height that slides across an image. The classifier decides whether the object sought is within the sliding window and therefore within the image. In addition, to detect objects at different scales, the process is repeated by resizing the image several times, resulting in what is called a "pyramid of images" [2]. This process, although efficient, is very slow because the classification, based on neural networks, must be repeated, sometimes, thousands of times for each image.

Recent advances in deep learning have provided more efficient object detection algorithms than the sliding window method. As a first alternative to the sliding window method, the

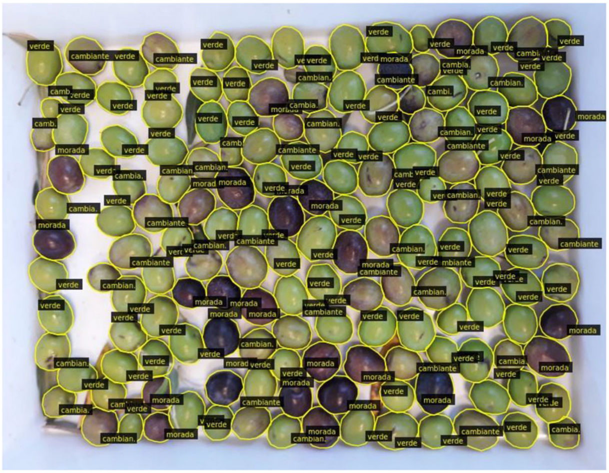

Fig. 3 "Ripeness" attribute labelling with VIA concept of R-CNN emerged in 2014 [9]. This method allows us to avoid the problem of selecting many different regions, as it uses a selective search algorithm [28] to extract just 2000 regions from the image, which acquire the name of "candidates" or "Regions of interest (ROIs)" and must be subsequently classified.

However, R-CNN continues to have a high computational cost. As alternatives, Fast R-CNN [8] and Faster R-CNN [25] emerged. In Fast R-CNN, the input image passes through the convolution operation only once and, thanks to special multi-scale pooling regions, proposals of arbitrary size can be processed by the fully connected layers. This allowed Fast-R-CNN to run much faster than R-CNN. However, in Fast-R-CNN the selective search algorithm must still be used, and this algorithm is computationally expensive. Faster R-CNN eliminates the use of the selective search algorithm, and a new separate neural network (the RPN network) is used to predict the candidates. Thanks to its efficiency in computing time, Faster R-CNN enables real-time object detection. In 2017, researchers at Facebook AI Research (FAIR) created an extension of Faster R-CNN called Mask R-CNN [12]. For a given image, Mask R-CNN, in addition to the class label and bounding box coordinates for each object, returns the object mask as well.

These algorithms were followed by other detection models that increase the speed at the expense of the quality of the proposals. Among the most outstanding algorithms are "You Only Look Once" (YOLO) [23] and SSD (Single Shot Multibox Detector) proposed by Liu et al. [18]. YOLO differs greatly from the algorithms seen before as it uses a single convolutional network that predicts the bounding boxes and the class probabilities for these boxes in a single pass. YOLO is a very fast algorithm that reaches speeds of up to 45 FPS (frames per second) but with the limitation to find small objects in images. Finally, in SSD the image only passes through the neural network once and it is composed of a first convolutional network followed by different convolutional layers that, at different scales, can predict the coordinates of the object and the class to which it belongs. SSD reaches speeds of up to 59 FPS.

As Faster R-CNN is considered the reference model for object detection thanks to the accuracy and robustness of its predictions, in this work, we have used Mask R-CNN, an evolution of Faster R-CNN, which is able to perform an image segmentation providing a pixel-wise mask for each object in the image.

2.3 Object detectors configuration

The objective of this work is to measure the accuracy of our object detector in the segmentation of each olive in the image according to the different attributes and IDs mentioned in Section 2.1. For this purpose, we have used an implementation of Mask R-CNN on Python 3, Keras and TensorFlow [1]. This implementation is based on Feature Pyramid Networks (FPN) [17]. To adapt the model to a particular problem we must specify the values of certain

Table 1 Different topologies for estimating the performance of detectors versus run time

TOPOLOGY	IMAGE_MIN_DIM	IMAGE_MAX_DIM	RPN_ANCHORS_SCALES
DIM6	128	192	15 17 20 23 25
DIM5	256	320	20 25 30 35 40
DIM4	384	448	30 35 40 45 50
DIM3	576	640	40 50 60 70 80
DIM2	768	832	40 55 70 85 100
DIM1	960	1024	55 75 95 105 125

parameters in a configuration file. These parameters refer, for example, to the number of classes (NUM_CLASSES), aspect ratio (RPN_ANCHOR_RATIOS) and size (RPN_ANCHOR_SCALES) of the anchors, dimensions of the input image (IMAGE_MIN_DIM and IMAGE_MAX_DIM), convolutional network model that performs the feature extraction (BACKBONE), etc.

In all the experiments we have used a ResNet-50 [11] as BACKBONE and a RPN_ANCHOR_RATIOS of 1:1 because the bounding boxes containing the objects to detect are approximately square.

As the goal is to deploy the object detector in a low-cost system and Mask R-CNN algorithm is very time consuming, especially if the dimension of the input image is very large, we have analysed the performance of the system based on this dimension. The dimension is set in the configuration file through parameters IMAGE_MIN_DIM and IMAGE_MAX_DIM. We have used the "square" resizing mode. In this mode, images are scaled up such that the small side is equal to IMAGE_MIN_DIM but ensuring that the scaling does not make the long side greater than IMAGE_MAX_DIM. Then, the image is padded with zeros to make it a square so multiple images can be put in one batch. In this implementation of Mask R-CNN, image dimension must be divisible by 2.[6] On the other hand, we have set the dimension of the anchors (RPN_ANCHOR_SCALES: length of square anchor side in pixels) based on the values of the dimensions of the input image. Consequently, we have completed simulations with the six topologies described in Table 1.

To train our model, we have applied transfer learning strategy [27]. It basically consists of assuming that a previously trained network, usually with many prototypes, can be used as a starting point to tackle another problem for which fewer patterns are available. So, we have used pre-trained weights for a Microsoft COCO dataset [16]. In the training process, we have only trained the last layers of the model, in particular the layers named *mrcnn_class_logits*, *mrcnn_mask*, *rpn_model*, *mrcnn_bbox* and *mrcnn_bbox_fc*. In the training process we used the early stopping technique and after the training we tried to improve the model by fine-tuning the weights. This was done by training all the layers of the network with a learning rate equal to one tenth of the original one and then re-training only the last layers, but this did not bring any significant improvement". The training of the models has been done on a GTX 1080 Ti GPU card with 8GB of RAM.

2.4 Metrics used for evaluation

Once our object detector has been trained, in the inference phase, the object detector identifies several regions of interest (ROIs) with a level of confidence (score) for each of them. Many of

Table 2 Results of the objects detectors with different topologies over the attribute 'object'

TOPOLOGY	Time (s)	TP	FP	FN	Precision	RECALL	F1-score	mAP
DIM6	0.14	893	8	28	0.991	0.970	0.980	0.971
DIM5	0.19	915	2	6	0.998	0.993	0.996	0.994
DIM4	0.31	918	2	3	0.998	0.997	0.997	0.997
DIM3	0.54	917	0	4	1	0.996	0.998	0.996
DIM2	0.86	919	0	**2**	1	0.998	0.999	0.998
DIM1	1.25	916	0	5	1	0.995	0.997	0.995

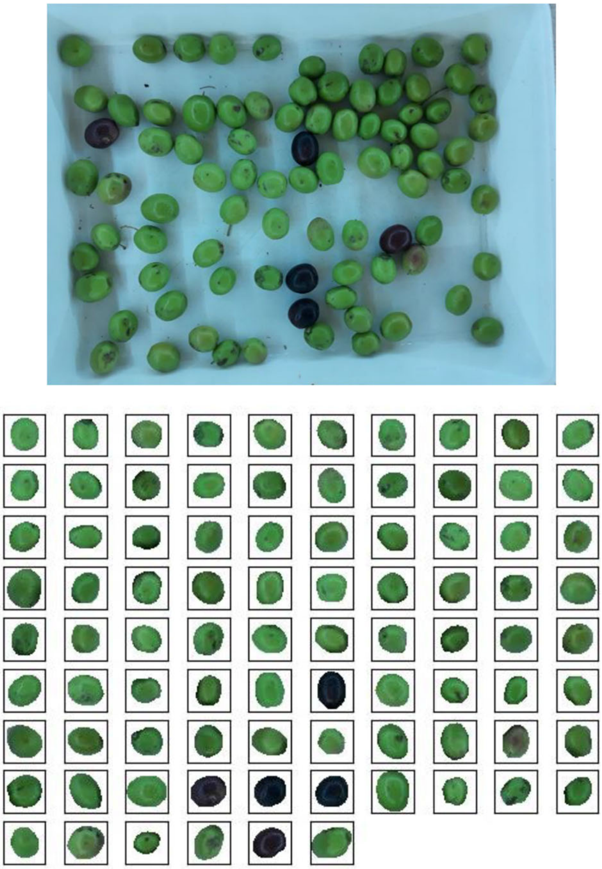

Fig. 4 Detected objects from one of the images of the test set with the DIM2 topology of the Table 2

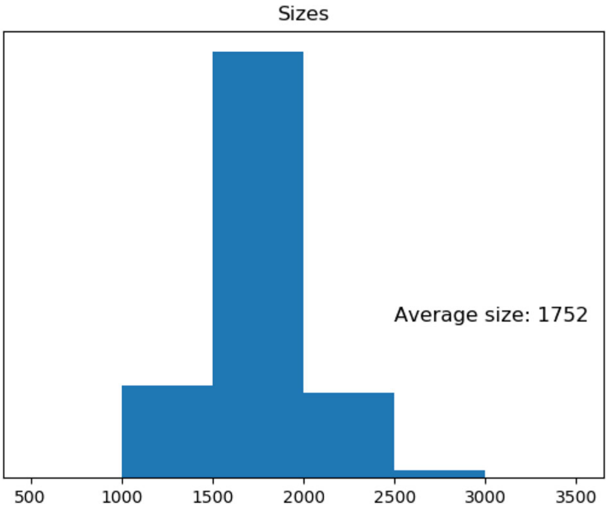

Fig. 5 Histogram with the sizes in pixels of the olive fruits in the image shown in Fig. 4

Table 3 Results of the objects detectors with different topologies over the attribute 'colour'

TOPOLOGY	Time (s)	TP	Notmatch	FP	FN	mF1	mAP
DIM6	0.14	768	66	9	87	0.865	0.909
DIM5	0.24	845	63	2	13	0.925	0.987
DIM4	0.33	861	52	0	8	0.939	**0.992**
DIM3	0.54	858	50	0	13	0.939	0.987
DIM2	0.87	862	41	0	18	0.947	0.982
DIM1	1.23	860	38	0	23	0.944	0.977

these candidates overlap, being representative of the same object and must be filtered by a Non-maximum Suppression algorithm (NMS). An NMS algorithm, based on the calculation of the Intersection-over-Unions (IoUs) and the value of a IoU NMS threshold (DETECTION_NMS_THRESHOLD = 0.3), selects the non-overlapping candidates with highest confidence level. The formal definition of IoU for two candidates of areas R1 and R2 is given by Eq. 1. Then, regions with a confidence level below a detection threshold (DETECTION_MIN_CONFIDENCE =0.7) are discarded. Now, with the final regions and with the ground truth of the images we must calculate metrics to evaluate the system performance over the test set. In this work, we have calculated four different metrics: precision, recall, F1-score and Average Precision (AP).

$$IoU(R1,R2) = \frac{|R1 \cap R2|}{|R1 \cup R2|} \tag{1}$$

To conclude that a candidate generated by our system has located an olive fruit, we must use an overlap metric with respect to the reference marks. This metric is, again, the IoU that is used to measure the accuracy of a detection candidate. Now the regions R1 and R2 of the Eq. 1 are both the area of the candidate and the area of a ground-truth mark, respectively. The "IoU detection threshold" may be different from that used by the NMS algorithm. In this case, being 1 the value reached when a candidate perfectly overlaps with the ground truth bounding box, a minimum value of 0.5 is usually considered a good object detection [6]. Consequently, a detection will be a true positive (TP) when its IoU with the reference mark exceeds the established IoU detection threshold. If a candidate does not reach a minimum overlap with any ground-truth mark it will be considered a false positive (FP). And finally, each non-localized olive fruit will be considered a false negative (FN). Using these markers, the usual metrics of

Table 4 Confusion matrix of the DIM4 topology detector of the Table 3

	DIM4	Predicted label				
		Green	Semi-ripe	Ripe	FN	Recall
True Label	**Green**	**729**	23	0	3	0.966
	Semi-ripe	21	**72**	4	4	0.713
	Ripe	0	4	**60**	1	0.923
	FP	0	0	0		
	Precision	0.972	0.727	0.938		
	F1-score	0.969	0.72	0.93		
	mF1		0.9388			

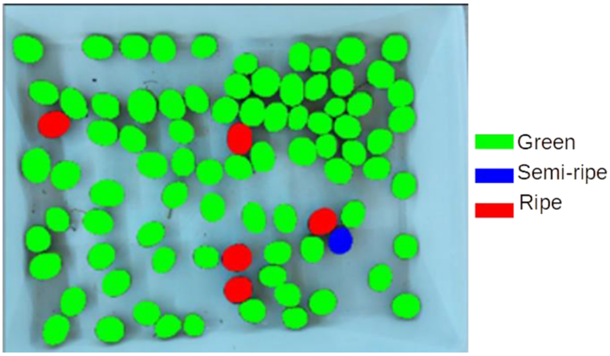

Fig. 6 Masks of the objects detected for the image in Fig. 4 with the network with the DIM4 topology of the Table 3

precision, recall and F1-score are defined in Eqs. 2 and 3 [15]. Precision informs about the ability of a classifier to identify relevant objects. Recall, on the other hand, measures the ability of the model to find all relevant cases. And finally, F1-score informs whether a model has been adjusted to favour precision over recall or vice versa. This parameter reaches a maximum value of 1 and decreases with decreasing precision or sensitivity.

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1\text{−}score = 2\frac{Precision\ x\ Recall}{Precision + Recall} \tag{3}$$

On the other hand, in multi-class object detection problems where, in addition to locating the object, it is necessary to associate it with its corresponding class, these values are calculated for each of the classes from the confusion matrix. Thus, precision for class C_i is calculated as the number of correctly predicted C_i out of all predicted C_i and recall for class C_i is the number of correctly predicted C_i out of the number of actual C_i. For unbalanced test sets, these parameters are calculated for each category and weighed against the number of items in each category.

Finally, the area under the precision-recall curve (AP) can be used as a single metric to summarize the performance of the object detection model. For a given IoU detection threshold, a model with high precision at all recall levels will have a high AP score. In a multi-class object detection task the mean Average Precision is used (mAP), where individual AP is averaged over all classes.

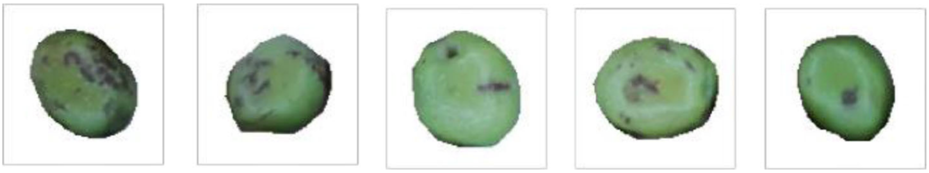

Fig. 7 Some prototypes for the “damaged” class

Table 5 Results of the objects detectors with different topologies over the attribute 'quality'

TOPOLOGY	Time (s)	TP	Notmatch	FP	FN	mF1	mAP
DIM6	0.13	492	264	10	165	0.581	0.826
DIM5	0.18	570	177	2	174	0.677	0.813
DIM4	0.31	571	155	1	195	0.693	0.792
DIM3	**0.60**	**711**	**177**	**1**	**33**	**0.785**	**0.967**
DIM2	0.81	691	107	0	123	0.798	0.869
DIM1	1.74	733	144	0	44	0.815	0.952

3 Results

3.1 Localization of olive fruits and estimation of their size

The objective of this experiment has been to segment each olive fruit in the image and from the mask obtained for each object in order to estimate its size according to the number of pixels occupied by the mask. In this case we used the attribute "object" and its unique ID "olive". In the literature this problem is called object localization problem since all objects belong to the same class.

Table 2 shows the execution time per image, on a GTX 1080 Ti GPU card, and the detection results (metrics defined in Section 2.3) on test images for the different topologies. In that case the images of the test set contain 921 olive fruits and the values of the F1-score and the mAP are very good for the first five topologies. The one with the best results is the DIM2 topology with a F1-score value of 0.999, a mAP value of 0.998 and an execution time of 0.86 s per image.

In Fig. 4 we can observe an image of the test set and the masks of the objects predicted by the detector with the DIM2 topology. In this case, the number of FNs and FPs are zero and all the 86 olives fruits are correctly detected.

In Fig. 5 we can see a histogram with the dimensions, in pixels, of the olive fruits in the image of Fig. 4. From this histogram we can calculate the average size of the olives, which is one of the fundamental parameters to be extracted during the "escandallo".

3.2 Detection of olive fruits maturity

For this experiment we have used the attribute "ripeness" which contains three IDs: "green", "semi-ripe" and "ripe". In this case, we have four classes including the "background" (no

Table 6 Confusion matrix of the network with the DIM3 topology of the Table 5

	DIM3	Predicted label					
		Damaged	Green	Semi-ripe	Ripe	FN	Recall
True Label	**Damaged**	**372**	68	10	0	11	0.807
	Green	62	**206**	7	0	16	0.708
	Semi-ripe	11	9	**73**	7	4	0.702
	Ripe	0	0	3	**60**	2	0.923
	FP	1	0	0	0		
	Precision	0.834	0.728	0.785	0.896		
	F1-score	0.820	0.718	0.741	0.909		
	mF1		0.785				

object of interest) class. It is an object detection problem, as objects must be localized and correctly associated with their corresponding class. Now, 921 olive fruits in the images of the test set are distributed as 755 greens, 101 semi-ripes and 65 ripes. In Table 3 classification results reached by the detectors with the different topologies are shown. We can observe a new column in the table named “notmatch” that counts those objects correctly localized but not associated to their corresponding class. Furthermore, the $F1_{\text{-score}}$ and AP values have been calculated for each of the classes and weighed against the number of items in each class obtaining the values of the mF1 (mean F1) and mAP (mean AP).

In this case, the DIM4 topology offers the best value of mAP, a good value of mF1 and an execution time four times smaller than DIM1 topology. In Table 4 the confusion matrix and the, Precision, Recall, FPs, FNs and $F1_{\text{-score}}$ for each class of the network with the DIM4 topology are shown.

Figure 6 shows the pixel-wise masks of the objects detected in the image shown in Fig. 4 by the network with the DIM4 topology of Table 3. Objects have been coloured according to the class to which they belong: green for the “green” olives, red for the “ripes” and blue for the “semi-ripes”.

3.3 Detection of defects in olive fruits

For this experiment, we have used the attribute “quality” with the IDs “damaged”, “green”,” semi-ripe” and “ripe” defined with the VIA software. In addition to detecting the state of maturity of the olive fruits, we also intend to determine the percentage of green fruits that are damaged by different causes like diseases, pest or damage caused in the harvesting process. In this case, of the 921 olive fruits contained in the images of the test set, 461 have been categorized as damaged, 291 as green, 104 as semi-ripe and, finally, 65 as ripe. In Fig. 7, we can see several samples of green olive prototypes for the “damaged” class.

The results of the object classifiers with the different topologies can be seen in Table 5. In Table 6 we show the confusion matrix of the network with the DIM3 topology which is the best in terms of mAP and shows an execution time three times smaller than DIM1 topology.

Figure 8 shows the pixel-wise masks of the objects detected in the image shown in Fig. 4 by the network with the DIM3 topology of the Table 5. Objects have been coloured according to the class to which they belong: yellow for the “damaged” olives, green for the “green”, red for the “ripes” and blue for the “semi-ripes”.

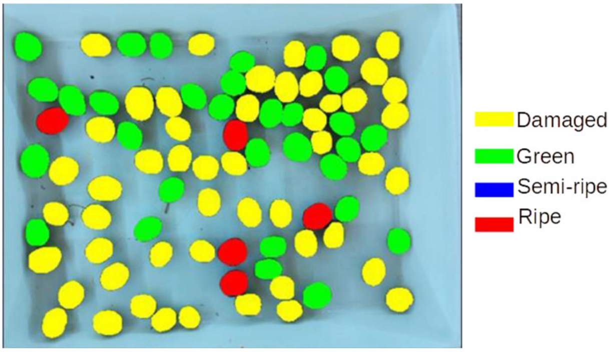

Fig. 8 Masks of the objects detected for the image in Fig. 4 with the network with the DIM3 topology of the Table 5

4 Discussion and conclusions

In this paper we have presented a system capable of localizing, with excellent accuracy, olive fruits in an image.

First, in Section 3.1, we have observed that 919 of 921 objects presented in the test set have been correctly extracted by the detector with the DIM2 topology (99.8%). From the results, we can deduce that the task of extracting the olives of an image, counting them, and inferring their sizes is practically solved even without strictly controlled environmental conditions.

Second, in addition to detecting the objects, when we have tried to classify them into different categories according to their state of maturity or quality of the fruit, the results have been less accurate. There are two reasons for the observed deterioration in the results: 1) the slight increase in false negatives and 2) the cases of confusion between neighbouring classes. However, values of mAP are close to 1, 0.992, for the network with the DIM4 topology of Table 3 and 0.967 for the network with the DIM3 topology of the Table 5. If we estimate the number of olive fruits localized and classified correctly by dividing the elements of the diagonal of the confusion matrix of Tables 4 and 6 by the number of olive fruits (921) we obtain success rates of 93.5% and 77.9%, respectively. It should be noted that, especially in the detection of damaged olives, the worsening of the results was to be expected. This is because in the prototype selection phase, it is sometimes difficult to select the threshold that separates a green olive from a semi-ripe olive, or to what extent a very small, or almost imperceptible defect means that an olive is considered damaged or not.

In future research, we will try and avoid the deterioration of detection results due to misclassifications. This would involve exploring the use of the object detector only for the localization of individual objects in the images followed by a convolutional network trained to individually classify each of the objects into the established classes. Furthemore, especially in the case of detection of defects in olives, some salient based method [13] will be tested for improving the accuracy of detection.

Furthermore, as we can see in Fig. 5, the size of the located objects is expressed in pixels. It is true that the size (area) of the olives could be easily deduced by considering the number of pixels extracted from the masks, the distance at which the photograph is taken and the parameters of the camera. However, traditionally, and possibly to facilitate the process of scanning, this size is expressed in weight (number of fruits per kilogram as described in the introduction). So, a correlation should be made between the number of pixels and the real sizes of the olives defined in the standards. To make this correlation we must wait for the next harvesting campaign.

Finally, the implementation of the system in a Raspberry Pi 4 model B 8G shows an execution times of approximately 40 seconds per image. We will adapt the algorithm to run on specific boards for machine learning and deep learning such as Jetson Nano and Intel Movidius Neural Compute Stick and thus achieve improvements in execution times.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Data availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Abdulla W (2017) Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. Retrieved from https://github.com/matterport/Mask_RCNN. Accessed 15 Feb 2023
2. Adelson EH, Anderson CH, Bergen JR, Burt PJ, Ogden JM (1984) Pyramid methods in image processing. RCA Engineer, 29-6 Nov/Dec 1984
3. Aquino A, Ponce JM, Andújar JM (2020) Identification of olive fruit, in intensive olive orchards, by means of its morphological structure using convolutional neural networks. Comput Electron Agric 176:105616. https://doi.org/10.1016/j.compag.2020.105616
4. Blasco J, Munera S, Aleixos N, Cubero S, Molto E (2017) Machine vision-based measurement systems for fruit and vegetable quality control in postharvest. Advances in Biochemical Engineering / Biotechnology, 161:71–99. https://doi.org/10.1007/10_2016_51
5. Braverman V (2016) Sliding window algorithms. In: Kao MY (eds) Encyclopedia of algorithms. Springer, New York, NY. https://doi.org/10.1007/978-1-4939-2864-4_797
6. Everingham M, Eslami SMA, Van Gool L, Williams CKI, Winn J, Zisserman A (2015) The pascal visual object classes challenge: a retrospective. Int J Comput Vis 111(1):98–136. https://doi.org/10.1007/s11263-014-0733-5
7. Gallardo-Caballero R, García-Orellana CJ, García-Manso A, González-Velasco HM, Tormo-Molina R, Macías-Macías M (2019) Precise pollen grain detection in bright field microscopy using deep learning techniques. Sensors 19:3583. https://doi.org/10.3390/s19163583
8. Girshick R (2015) Fast R-CNN. IEEE international conference on computer vision, pp 1440–1448. https://doi.org/10.1109/ICCV.2015.169
9. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. IEEE conference on computer vision and pattern recognition, pp 580–587. https://doi.org/10.1109/CVPR.2014.81
10. Guzmán E, Baeten V, Pierna JA, García-Mesa JA (2015) Determination of the olive maturity index of intact fruits using image analysis. J Food Sci Technol 52(3):1462–1470. https://doi.org/10.1007/s13197-013-1123-7
11. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. IEEE conference on computer vision and pattern recognition (CVPR), p. 770–778. https://doi.org/10.1109/CVPR.2016.90
12. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask R-CNN. IEEE international conference on computer vision (ICCV), pp 2980–2988. https://doi.org/10.1109/ICCV.2017.322
13. Jia S, Zhang Y (2018) Saliency-based deep convolutional neural network for no-reference image quality assessment. Multimed Tools Appl 77:14859–14872. https://doi.org/10.1007/s11042-017-5070-6
14. Kaur H, Sawhney BK, Jawandha SK (2018) Evaluation of plum fruit maturity by image processing techniques. J Food Sci Technol 55:3008–3015. https://doi.org/10.1007/s13197-018-3220-0
15. Koirala A, Walsh KB, 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
16. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollar P, Zitnick CL (2014) Microsoft COCO: common objects in context. The 13th European conference on computer vision (ECCV), pp 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

17. Lin T, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. IEEE conference on computer vision and pattern recognition (CVPR), pp 936-944. https://doi.org/10.1109/CVPR.2017.106
18. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C, Berg AC (2016) SSD: single shot multibox detector. In: Leibe B, Matas J, Sebe N, Welling M (eds) Computer vision – ECCV 2016. Lecture notes in computer science, 9905, 21–37. Springer, Cham. https://doi.org/10.1007/978-3-319-46448-0_2
19. Munera S, Hernández F, Aleixos N, Cubero S, Blasco J (2019) Maturity monitoring of intact fruit and arils of pomegranate cv. 'Mollar de Elche' using machine vision and chemometrics. Postharvest Biol Technol 156. https://doi.org/10.1016/j.postharvbio.2019.110936
20. Piedad E, Larada JI, Pojas GJ, Ferrer LVV (2018) Postharvest classification of banana (Musa acuminata) using tier-based machine learning. Postharvest Biol Technol 145:93–100. https://doi.org/10.1016/j.postharvbio.2018.06.004
21. Ponce JM, Aquino A, Millán B, Andújar JM (2018) Olive-fruit mass and size estimation using image analysis and feature modeling. Sensors 18(9):2930. https://doi.org/10.3390/s18092930
22. Ramos PJ, Avendaño J, Prieto FA (2018) Measurement of the ripening rate of coffee branches by using 3D images in outdoor environments. Comput Ind 99:83–95. https://doi.org/10.1016/j.compind.2018.03.024
23. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. IEEE conference on computer vision and pattern recognition (CVPR), pp 779-788. https://doi.org/10.1109/CVPR.2016.91
24. Rejano L, Montaño A, Casado FJ, Sánchez AH, De Castro A (2010) Chapter 1 - table olives: varieties and variations. Olives and Olive Oil in Health and Disease Prevention, Academic Press, 5–15. https://doi.org/10.1016/B978-0-12-374420-3.00001-2
25. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
26. Riquelme MT, Barreiro P, Ruiz-Altisent M, Valero C (2007) Olive classification according to external damage using image analysis. J Food Eng 87(3):371–379. https://doi.org/10.1016/j.jfoodeng.2007.12.018
27. Tan C, Sun F, Kong T, Zhang W, Yang C, Liu C (2018) A survey on deep transfer learning. The 27th international conference on artificial neural networks (ICANN 2018), 270–279. https://doi.org/10.1007/978-3-030-01424-7_27
28. Uijlings JRR, van de Sande KEA, Gevers T, Smeulders AWM (2013) Selective search for object recognition. Int J Comput Vis 104(2):154–171. https://doi.org/10.1007/s11263-013-0620-5

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.