Automated classification of olive fruit for enhanced olive oil production using computer vision

Erion-Vasilis Pikoulis[a], Kostas Blekos[a], and Dimitrios Kosmopoulos[a]

[a] *Computer Engineering and Informatics Department, University of Patras, Greece*

ABSTRACT

The olive oil industry plays a significant role in the global agricultural economy, with the quality of olive oil greatly dependent on the quality and ripeness of the olives used in its production. Accurate and efficient sorting and classification of olive fruit are crucial steps in optimizing olive oil yield and quality. In this scientific project, we propose a novel approach to automate the classification of olive fruit based on their ripeness and quality using computer vision techniques. The visual system is composed of a segmentation and classification deep network, based on YOLO architecture. In practice despite the processing unexpected foreign objects may be present as well (e.g. leaves, twigs etc), which may lead to erroneous classification to one of the existing classes. A classification problem is therefore defined. The experimental results validate the utility of the approach with high classification accuracy based on expert annotation and demonstrate high detection rates for outlier objects. The speed of the system ensures a high production throughput.

Keywords: automatic olive fruit classification, olive oil production, object detection

1. INTRODUCTION

Extra virgin olive oil (EVOO) is one of the main ingredients of the Mediterranean diet thanks to its organoleptic and healthy properties.[1] 82% of the world production of EVOO comes from the European Mediterranean countries. Indeed, the Mediterranean region, thanks to its dry and subtropical climate, constitutes a favorable growth zone for the olive tree. EVOO is produced from the fruit of the olive tree (Olea europaea), through mechanical or natural processes that include washing, decantation, centrifugation and filtration. An EVOO to be defined as such must be produced through these processes and have a number of organoleptic characteristics,[2] which are due to the presence of two fractions, one containing the main components (triglycerides, diglycerides and free fatty acids) and the other containing the secondary components (pigments, tocopherols and phenolic compounds).[3]

It is important to underline that the quality of EVOO is strongly dependent on the condition of the olive fruit, which is influenced by several factors, such as the ripening process, the atmospheric and agricultural conditions, the genetic control.[4] As such, establishing a rigorous monitoring and selection mechanism ensuring the health and suitability of the fruits that enter the extraction process, is one of the key factors determining the quality of the final EVOO product. Today, this selection process can be accomplished using machines capable of selecting large batches of fruit based on real-time image analysis, enabling an objective and fast selection that is impossible to perform by hand. However, although optical sorting machines are widely used to select different types of fruits and vegetables, olive sorting is an still an emerging field (in the EVOO production chain) albeit a very promising and rewarding one.

In this work we present a methodology on olive detection and classification in a production setting which involves a conveyor belt and custom illumination. The setting cannot exclude the presence of outlier objects like leaves, small stones, twigs, etc., which can have unexpected visual features. Therefore we model the problem as an open visual object classification problem. To this end we use an off-the-shelf object detector like YOLO

Further author information: (Send correspondence to Dimitrios Kosmopoulos)

D. Kosmopoulos: E-mail: dkosmo@upatras.gr

K. Blekos: E-mail: mplekos@upatras.gr

E. V. Pikoulis.: Email: pikoulis@ceid.upatras.gr

and use an openmax approach for the classification step. We provide qualitative and quantitative results of the proposed approach.

In the following section we present the related works. In section 3 we present the overall system and its architecture. In section 4 we give the methodology of our classification framework. In section 4.1 we describe the dataset. In section 5 we present the experimental results, while section 6 concludes the paper.

2. RELATED WORK

The simplest methods do color segmentation and classify the olive fruit based on that. For example Violino et al[5] use a color - based classifier is used to classify the fruit into green and black classes. It was found from the analyses that the oils obtained show significant differences both chemically and through expert panel tests. Puerto et al[6] have classified olives, collected on the plant or from the ground, following the roughness of the surface and the color of the fruits. Also, Babanatis et al[7,8] developed an image analysis classifier to distinguish olives based on their color.

In recent years, DNN models have contributed significantly to the performance improvement of image-based object detection (see Zou et al[9]). Works concerning DNN-based 2D detectors can be broadly categorized into two-stage and one-stage approaches (Yurtsever et al[10]). Detectors following the two-stage approach first generate region proposals on the input image and then assess each region regarding the presence of one or multiple objects and the class each of them belongs to. On the other hand, single-stage object detectors produce directly both the location and the class of each object in the input image. Although two-stage detectors usually perform better,[9] single-stage detectors such as the Single Shot MultiBox Detector (SSD) by Liu et al,[11] and the You Only Look Once (YOLO) detector by Redmon et al,[12] which are generally preferred due to their lower computational and storage requirements.

As expected, DNN-based classifiers and object detectors have also been used for olive fruit classification. In the first case, In the work of Figorilli et al.,[13] the classification of olive fruits into five classes of quality using a CNN has been investigated that employs the well-known Alexnet as a backbone network for automatic feature extraction when the input is RGB images. The work of Saedi et. al.[14] investigates the problem of classifying olive fruits into five ripening stages by employing transfer learning to a pre-trained (using the ImageNet dataset) version of the Xception model. Focusing on lightweight neural network models that can be deployed on resource-constraint devices,[15] proposes a custom CNN architecture that can classify among three varieties of olive fruits and two classes of quality. In the case of employing object detectors, YOLO detector was recently used to support the production of high-quality EVOO by automatically and quickly classifying olive lots into different quality classes, for both oil production and table olives (see e.g., Salvucci et al[16]). A similar approach has been studied in[17] in which the latest two versions of Yolo (v7 and v8) have been considered for counting.

Many studies found in the literature concern the evaluation of the image analysis algorithm for the selection of olives in rather simplistic settings. On the contrary a realistic selection test may involve non-uniform background, self-occlusions and foreign objects that may interfere in the process. Very few works have examined realistic olive selection systems, even on a small laboratory scale. Some investigated the selection in the laboratory of small quantities of olives at low speed by separating one olive at a time.

3. AUTOMATIC OLIVE-FRUIT SORTING SYSTEM

The presented image-based quality analysis solution will eventually be integrated to an automatic olive-fruit sorting machine, capable of sorting olive batches in real-time. The machine is composed of a conveyor belt with attached specialized scoops, with compartments capable of carrying a single olives at a time (please see Fig. 1 for a depiction of the sorting machine under development). A high-resolution camera and lighting set-up is currently being fitted to the conveyor belt, while the final prototype will also leverage an electronic control unit (ECU) and a processing unit, as well as a row of servo-based actuators (one for each olive compartment) able to physically separate the olives as they reach the end of the belt, based on the decisions of the automated image-based quality control system. In its operational phase, the camera setup will be timed to capture each scoop as it passes through it. Afterwards, the image will be fed to the DNN based object detector with the goal of identifying and classifying the contents (i.e. healthy or damaged olive fruit) of each compartment. The

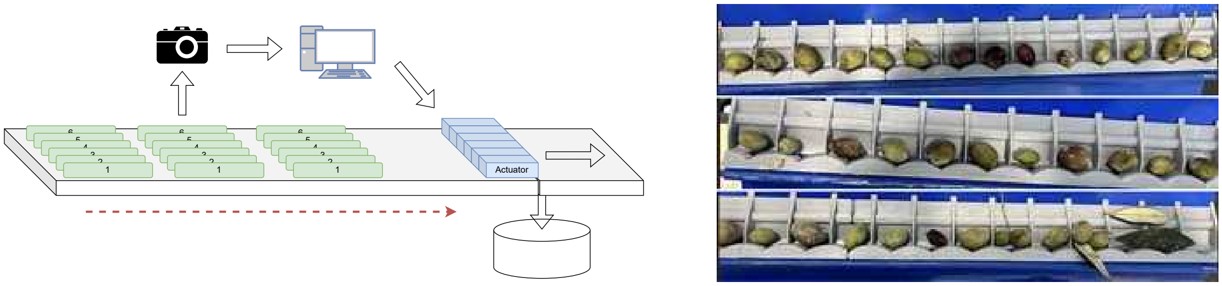

Figure 1. A schematic diagram of the sorting machine under development (left) and close-up examples (right).

Figure 2. Instances from the “bad” (upper row), “good” (middle row), and “leaf” (lower row) classes defined in the used dataset.

decisions will be then fed in time to the actuators, so that, by the time each scoops reaches the actuators, they latter have been correctly configured allowing the healthy ones to pass through for further processing, while redirecting the damaged ones to a different container.

4. METHODOLOGY

In this paper, we are presenting preliminary results of the olive-fruit sorting system, focusing on the image-based classification tool. For the purposes of the study we collected a a dataset containing both healthy and damaged olive fruits, as well as extraneous objects such as leaves, twigs, etc.

4.1 Dataset

A new dataset of olive-fruit images was created from scratch for the purposes of this study. The data consists of images recorded in realistic conditions using the sorting system under development. Specifically, since the feeding mechanism and camera/lighting setup are not yet in place, the olive batches were placed by hand on the conveyor belt, and then photographed using a smartphone. A total of 152 captures were taken for the dataset. The used olives were a mix of good and damaged fruits, from the Koroneiki cultivar, including also extraneous objects (mostly leaves) for testing purposes. Koroneiki fruits are small, but have a high quality oil yield, and are utilized for EVOO extraction during their green phase of maturity.

The captured images were subsequently hand-annotated using the roboflow web-based tools,[18] by drawing a bounding box surrounding each object of interest and assign a label to the selected object (please see Fig. 3 for examples of the dataset). For our purposes, we discriminated between two olive classes, namely “good” (green olives with smooth texture and little to no damage), “bad” (dented, damaged, and/or over-matured), and “leaf”. The last category was added so that the machine learns to explicitly discriminate between olive-fruit and leaves. A total of 1941 objects were hand-annotated, consisting of 1015 “bad” olives, 728 “good” olives, and 198 “leaves”. Object examples from the three used classes are depicted in Fig 2. The annotated dataset is available online at the roboflow universe[19]

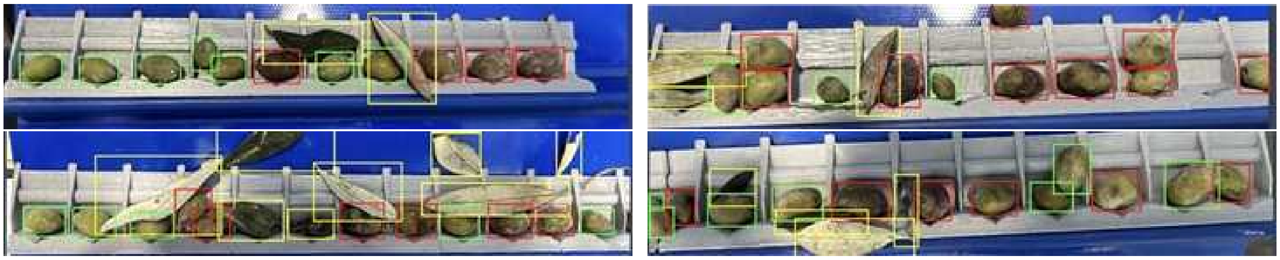

Figure 3. Annotated examples from the used dataset depicting bounding boxes for good (green) and bad (red) olives, as well as leafs (yellow).

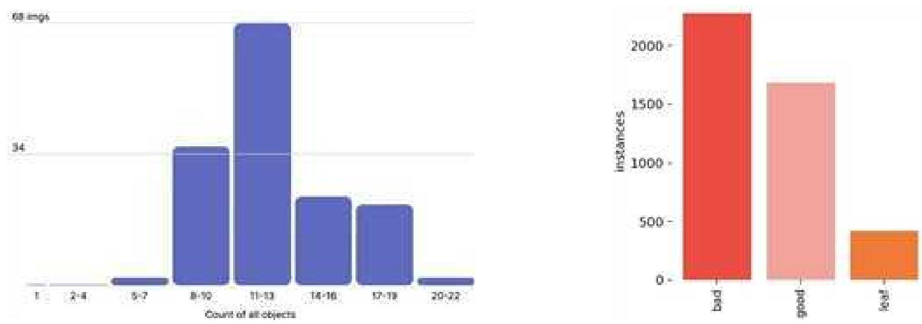

Figure 4. Histograms of objects per image (left), and total number of instances per class (right) in the trainng dataset.

4.2 DNN-based sorting mechanism

In this work, we utilize DNN-based object detection and classification, from the families of YOLO detectors and Residual Network (ResNet) classifiers, respectively.

YOLO was introduced to the computer vision community in 2016 through a paper by Joseph Redmon et al.[12] The paper redefined object detection by presenting it as a single-pass regression problem, starting from image pixels and progressing to bounding box and class probabilities. This approach, grounded in the 'unified' concept, facilitated the simultaneous prediction of multiple bounding boxes and class probabilities, thereby enhancing both speed and accuracy.[20] Subsequent revisions introduced further enhancements to the original architecture, drawing from advanced detection technologies available at the time (e.g., anchors[21] for YOLOv2,[22] Residual Net[23] for YOLOv3,[24] etc.) Since its inception, the YOLO family has experienced rapid expansion, culminating in the latest addition, the Ultralytics YOLOv8,[25] which is utilized in this paper. The most important new feature of YOLOv8 compared to its predecessors comes in the form of anchor-free detection[26] (meaning that it predicts directly the center of an object instead of the offset from a known anchor box), which reduces significantly the number of box predictions, thus speeding up Non-Maximum Suppression (NMS), a complicated post processing step that sifts through candidate detections after inference. For more details regarding the YOLO family of detectors, please see the work of Terven and Cordova-Esparza.[27]

On the other hand, ResNet[23] classifiers are based on the concept of residual learning. Its building block comprises of two consecutive 3×3 conv layers, with the block's output being summed to its input using a "bypass" connection (hence, the block is required to only learn the residual representation). The ResNet18 model used in this work represents a scaled-down version of bigger ResNet variants (e.g. ResNet34, ResNet50, etc.), consisting of 8 such blocks (plus an input conv layer and an fully connected layer), for a total of roughly 1.1×10^7 paramters and 1.8 GFLOPs.

5. EXPERIMENTAL RESULTS

In this section we present two sets of experimental results, each focusing on a different scenario regarding the automated olive-fruit sorting task. In the first experiment we address simultaneous detection and classification utilizing a YOLO-based detector, representing a scenario where the olives locations in the image are unknown. In the second case, we consider the potential olive locations as known (which represents a scenario more closely

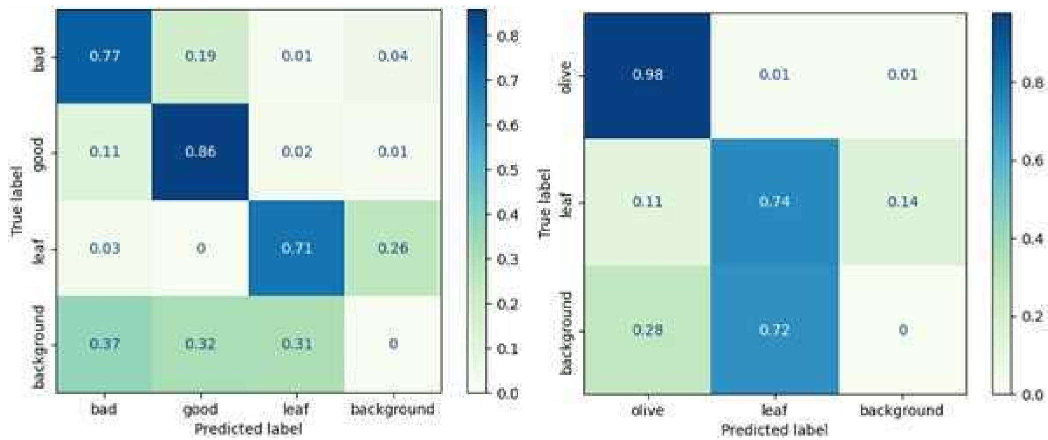

Figure 5. Normalized confusion matrices for the 3-class (left), and 2-class (right) detectors used in Experiment I.

related to the sorting mechanism depicted in Fig. 1) and focus on the classification between acceptable and non-acceptable olive fruits.

5.1 Experiment I: YOLO-based detection

In the first experiment, we use the annotated dataset to train an object detector to simultaneously locate the objects and classify them into the three predefined classes, i.e. {bad, good, leaf}. For this purpose we utilize a pre-trained version of the Ultralytics YOLOv8,[25] using specifically the lightweight 'YOLOv8n' variant (225 layers, 3.2×10^6 parameters, 8.9 GFLOPs). In a transfer learning fashion, we retain the backbone (feature-extraction) and re-configure the detection head for a 3-class detection task.

The newly formed model was then fine-tuned (re-trained) for 30 epochs using the annotated data presented inf Section 4.1. For training purposes, the dataset was split into train, validate and test sets, with a percentage of roughly 75% (114 images), 15% (22 images), 10% (15 images). The training set was further augmented with flipped and rotated images in the range $[-15°, 15°]$, for a total of 342 images ($3 \times$ increase). The distribution of objects and labels in the training dataset are depicted in Fig 4. The performance of the fine-tuned model is summarized by the confusion matrix shown in Fig. 5 (left), while detection instances using the available test images are presented in Fig.6.

As it can be seen in Fig. 5 (and is also evident by the examples of Fig. 6), the 3-class detector demonstrates a generally satisfactory performance reaching an accuracy level of 86% for the class of "good" olives, while falling slightly behind on the "bad" (olive) and "leaf" classes, with an accuracy of 77% and 71%, respectively. This behavior can be attributed to the fact that, on the one hand the "good" (olive) class is better defined than the "bad" (olive) one (since all "good" olive-fruits are expected to exhibit very similar features, which cannot be said for the class of "bad" olives defined in this work), and on the other, "leaf" was the less populated class in the dataset.

To gain further insight, we unified the "bad" and "good" olive classes, and reconfigured the YOLO-based detector for a 2-class "olive" vs "non-olive" (i.e., "leaf") task. The rest of fine-tuning procedure was identical to the one followed for the 3-class detector. The results shown in Fig. 5 (right) demonstrate the impressive performance of the YOLO-based detector from a purely (olive) detection standpoint, reaching an accuracy of "98%" regarding olive detection, despite the relatively limited training dataset.

5.2 Experiment II: Olive-fruit classification

In this experiment we focus purely on the classification aspect of the olive-fruit sorting problem, which, for the automated mechanism that is being developed, represents the more challenging of the two involved tasks (with localization being the other), taking into account that locations of olive-fruits in the acquired images is expected to be known with a relatively high degree of accuracy (given the specialized scoops and the fixed camera position).

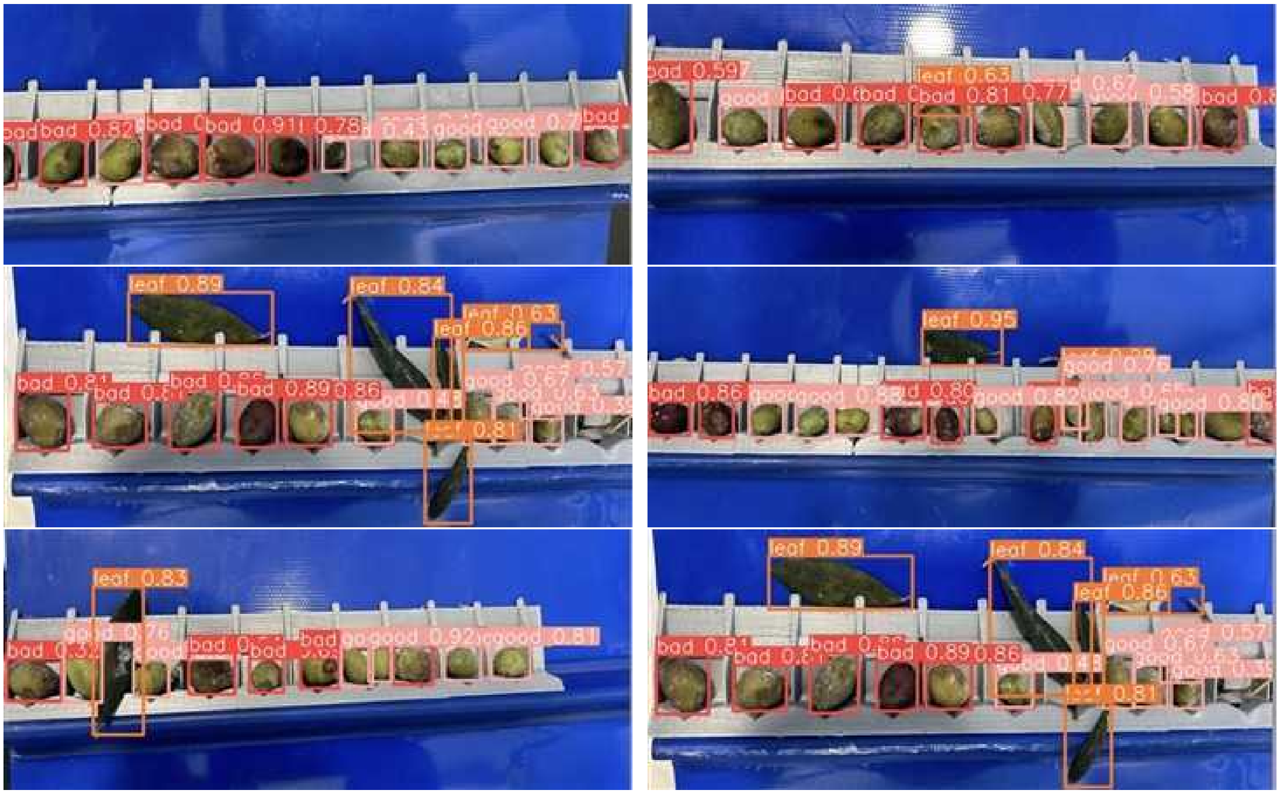

Figure 6. Automatic Olive-fruit detection and classification examples from the YOLO-based 3-class detector of Experiment I.

To this end, we used a ResNet18-based classifier (originally trained on the ImageNet image classification dataset[28]), replacing the 1000-way classification layer with a 2-way "bad" vs "good" layer. For training purposes we utilized the olive bounding boxes defined in the dataset described in Sec. 4.1, obtaining a 2-way classification dataset of 1015 and 728 images for the "bad" and "good" olive classes respectively. 50 images from each class were used for validation and 50 for testing purposes, with the rest being used for training. The ResNet18-based 2-way classifier was re-trained for 20 epochs, reaching an accuracy of roughly 88% for the classification task. This can be considered a very promising performance taking into account the relatively limited dataset, and the challenge posed regarding the discrimination between acceptable and non-acceptable olive-fruits, as described also in Experiment I.

Concluding this section, we note that all experiments were conducted on a PC with an Intel i5 processor, 16 GB RAM, and an Nvidia GeForce GTX 1050 Ti GPU (4GB RAM). The inference times for the YOLO-based detector and ResNet-based classifier were approximately 15 ms and 8 ms (or, 67 fps and 125 fps), respectively, which is more than adequate for the needs of the sorting mechanism under developement.

6. CONCLUSIONS

We have presented a system for sorting and classification of olive fruit in order to support extra virgin olive oil production. The olive fruit are sorted based on their ripeness and quality using a YOLO-based pipeline, which is accurate and efficient. The foreign objects that may be present are classified as such, resulting in only minor influence of the olive sorting. The experimental results validate the utility of the approach and the system speed of the system supports a high production throughput.

The proposed system in financially viable due of its very simple and cost-efficient components. In its current form the system is at the functional prototype level. In the near future we will study the behavior of the system in a fully automated fashion with large volumes of olive fruit.

ACKNOWLEDGMENT

This research has been funded by the Greek Ministry of Agricultural Development and Food, under Sub-Measure 16.1 – 16.2 “Establishment and Operation of Business Groups of the European Innovation Partnership for Productivity and the Sustainability Of Agriculture”, Action 2, project number M16SYN2-00229.

REFERENCES

[1] Violino, S., Pallottino, F., Sperandio, G., Figorilli, S., Antonucci, F., Ioannoni, V., Fappiano, D., and Costa, C., “Are the innovative electronic labels for extra virgin olive oil sustainable, traceable, and accepted by consumers?,” *Foods* **8**(11), 529 (2019).

[2] Garcia-Oliveira, P., Jimenez-Lopez, C., Lourenço-Lopes, C., Chamorro, F., Pereira, A. G., Carrera-Casais, A., Fraga-Corral, M., Carpena, M., Simal-Gandara, J., and Prieto, M. A., “Evolution of flavors in extra virgin olive oil shelf-life,” *Antioxidants* **10**(3), 368 (2021).

[3] Bengana, M., Bakhouche, A., Lozano-Sánchez, J., Amir, Y., Youyou, A., Segura-Carretero, A., and Fernández-Gutiérrez, A., “Influence of olive ripeness on chemical properties and phenolic composition of chemlal extra-virgin olive oil,” *Food Research International* **54**(2), 1868–1875 (2013).

[4] Manganiello, R., Pagano, M., Nucciarelli, D., Ciccoritti, R., Tomasone, R., Di Serio, M. G., Giansante, L., Del Re, P., Servili, M., and Veneziani, G., “Effects of ultrasound technology on the qualitative properties of italian extra virgin olive oil,” *Foods* **10**(11), 2884 (2021).

[5] Violino, S., Moscovini, L., Costa, C., Re, P. D., Giansante, L., Toscano, P., Tocci, F., Vasta, S., Manganiello, R., Ortenzi, L., and Pallottino, F., “Superior evoo quality production: An rgb sorting machine for olive classification,” *Foods* **11**(18) (2022).

[6] Puerto, D. A., Martínez Gila, D. M., Gámez García, J., and Gómez Ortega, J., “Sorting olive batches for the milling process using image processing,” *Sensors* **15**(7), 15738–15754 (2015).

[7] Babanatis-Merce, R. M., Babanatsas, T., Glavan, D. O., Glavan, A. I., Bucevschi, A., and Gaspar, M. C., “Automated tool for olive color recognition in sorting system development,” *Acta Technica Napocensis-Series: Applied Mathematics, Mechanics, and Engineering* **64**(1) (2021).

[8] Babanatis-Merce, R. M., Babanatsas, T., Glavan, D. O., Mircea, R., Glavan, A. I., Bucevschi, A., and Gaspar, M. C., “Programming optical sensors to increase performance of olive sorting system,” *Acta Technica Napocensis-Series: Applied Mathematics, Mechanics, and Engineering* **64**(1) (2021).

[9] Zou, Z., Chen, K., Shi, Z., Guo, Y., and Ye, J., “Object detection in 20 years: A survey,” *Proceedings of the IEEE* (2023).

[10] Yurtsever, E., Lambert, J., Carballo, A., and Takeda, K., “A survey of autonomous driving: Common practices and emerging technologies,” *IEEE Access* **8**, 58443–58469 (2020).

[11] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C., “Ssd: Single shot multibox detector,” in [*Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*], 21–37, Springer (2016).

[12] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., “You only look once: Unified, real-time object detection,” in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*], 779–788 (2016).

[13] Figorilli, S., Violino, S., Moscovini, L., Ortenzi, L., Salvucci, G., Vasta, S., Tocci, F., Costa, C., Toscano, P., and Pallottino, F., “Olive fruit selection through ai algorithms and rgb imaging,” *Foods* **11**(21), 3391 (2022).

[14] Saedi, S. I. and Rezaei, M., “A modified xception deep learning model for automatic sorting of olives based on ripening stages,” *Inventions* **9**(1), 6 (2023).

[15] Hayajneh, A. M., Batayneh, S., Alzoubi, E., and Alwedyan, M., “Tinyml olive fruit variety classification by means of convolutional neural networks on iot edge devices,” *AgriEngineering* **5**(4), 2266–2283 (2023).

[16] Salvucci, G., Pallottino, F., De Laurentiis, L., Del Frate, F., Manganiello, R., Tocci, F., Vasta, S., Figorilli, S., Bassotti, B., Violino, S., et al., “Fast olive quality assessment through rgb images and advanced convolutional neural network modeling,” *European Food Research and Technology* **248**(5), 1395–1405 (2022).

[17] Mendes, J., Lima, J., Costa, L. A., Rodrigues, N., Leitão, P., and Pereira, A. I., "An artificial intelligence-based method to identify the stage of maturation in olive oil mills," in [*International Conference on Optimization, Learning Algorithms and Applications*], 63–77, Springer (2023).

[18] "Roboflow." https://roboflow.com.

[19] "Olive classification dataset." https://universe.roboflow.com/olive-classification-gxzvf/olive-detection-s9rql.

[20] Jiang, P., Ergu, D., Liu, F., Cai, Y., and Ma, B., "A review of yolo algorithm developments," *Procedia computer science* **199**, 1066–1073 (2022).

[21] Ren, S., He, K., Girshick, R., and Sun, J., "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems* **28** (2015).

[22] Redmon, J. and Farhadi, A., "Yolo9000: better, faster, stronger," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*], 7263–7271 (2017).

[23] He, K., Zhang, X., Ren, S., and Sun, J., "Deep residual learning for image recognition," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*], 770–778 (2016).

[24] Redmon, J. and Farhadi, A., "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767* (2018).

[25] "Olive classification dataset." https://github.com/ultralytics/ultralytics.

[26] Law, H. and Deng, J., "Cornernet: Detecting objects as paired keypoints," in [*Proceedings of the European Conference on Computer Vision (ECCV)*], 734–750 (2018).

[27] Terven, J. and Cordova-Esparza, D., "A comprehensive review of yolo: From yolov1 to yolov8 and beyond," *arXiv preprint arXiv:2304.00501* (2023).

[28] Russakovsky et al., O., "Imagenet large scale visual recognition challenge," *IJCV* **115**(3), 211–252 (2015).