Tomato Detection Using Deep Learning for Robotics Application

Tiago Cerveira Padilha[1,2], Germano Moreira[4], Sandro Augusto Magalhães[1,3], Filipe Neves dos Santos[1(✉)], Mário Cunha[1,4], and Miguel Oliveira[2]

[1] INESC TEC - Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência, Campus da FEUP, Rua Dr. Roberto Frias, s/n 4200-465, Porto, Portugal
{tiago.padilha,sandro.a.magalhaes,fbsantos}@inesctec.pt

[2] Department of Mechanical Engineering, University of Aveiro, 3810 Aveiro, Portugal
{tiagopadilha,mriem}@ua.pt

[3] Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias, s/n 4200-465, Porto, Portugal
sandro.magalhaes@fe.up.pt

[4] Faculty of Sciences, University of Porto, Rua do Campo Alegre, s/n 4169-007, Porto, Portugal
{up201608269,mccunha}@fc.up.pt

Abstract. The importance of agriculture and the production of fruits and vegetables has stood out mainly over the past few years, especially for the benefits for our health. In 2021, in the international year of fruit and vegetables, it is important to encourage innovation and evolution in this area, with the needs surrounding the different processes of the different cultures. This paper compares the performance between two datasets for robotics fruit harvesting using four deep learning object detection models: YOLOv4, SSD ResNet 50, SSD Inception v2, SSD MobileNet v2. This work aims to benchmark the Open Images Dataset v6 (OIDv6) against an acquired dataset inside a tomatoes greenhouse for tomato detection in agricultural environments, using a test dataset with acquired non augmented images. The results highlight the benefit of using self-acquired datasets for the detection of tomatoes because the state-of-the-art datasets, as OIDv6, lack some relevant characteristics of the fruits in the agricultural environment, as the shape and the color. Detections in greenhouses environments differ greatly from the data inside the OIDv6, which has fewer annotations per image and the tomato is generally riped (reddish). Standing out in the use of our tomato dataset, YOLOv4 stood out with a precision of 91%. The tomato dataset was augmented and is publicly available (See https://rdm.inesctec.pt/ and https://rdm.inesctec.pt/dataset/ii-2021-001).

Keywords: Fruit detection · Machine learning · Computer vision · Agricultural robotics · Harvesting robotics

G. Marreiros et al. (Eds.): EPIA 2021, LNAI 12981, pp. 27–38, 2021.
https://doi.org/10.1007/978-3-030-86230-5_3

1 Introduction

The importance and benefits of fruit consumption are known to many and its production on a global scale has been a priority in the capacity for increased production. To ensure the conditions recommended by World Health Organization (WHO) [4], each person should consume at least 400 g of fruit and vegetables per day. The world production of fruits and vegetables in the year 2000 represented 306 g per day, in 2017 the production already represented 390 g [2]. An analysis of the growth of fruit primary production between 2000 and 2019, already represents an increase of approximately 65% according to FAOSTAT [1].

The various arduous tasks that sometimes the different stages of fruit cultivation represent, translate into a shortage of capable and specialized labor. This shortage of human resources is reflected in the difficult steps required to be taken in an agricultural process, from knowledge in analyzing products during their growth, such as harvesting, which in addition to being a stressful task is also time-consuming. The multiple problems associated with agriculture have opened the door to new technological solutions, including inspection and visual detection of fruits.

The implementation of automation in agricultural processes has been one of the most interesting solutions for companies looking to reduce costs and increase productivity. Robotic solutions have evolved and are increasingly suitable for environments in nature. They are usually composed of cameras and other sensors for the acquisition of images and detection of objects in real-time.

The visual detection of products in nature implies extra care in the unpredictability of events. Nature is unpredictable and an example of this is the heterogeneous characteristics that the same fruit can take, with respect to shape, color, size, branches, leaves, stems, as well as factors of variation in natural lighting.

Choosing the best descriptor for detecting or classifying a fruit is often a complex task, especially if we use traditional techniques. Deep Learning (DL) is based on non-linear models with a high capacity for learning data characteristics. The use of DL is a better approach in image processing than just using traditional methods. Computer image processing is one of the areas with the greatest application of artificial intelligence (AI) [16], but the quantity and quality of the dataset are essential for obtaining good results. In this context, the tomato dataset used presented Magalhães et al. [14], with improvements to annotations as well as the respective increase in the dataset.

The main goal of this study is to compare the training of DL model approaches using benchmark datasets against specific datasets. We can find different public datasets, which can be very useful for quick tests, training, and validating DL models. This approach can save several hours of acquiring and preparing a dataset, but we want to understand if it's enough to achieve the best results.

The following sections intend to illustrate how this research was conducted. Section 2 provides a review of the state of the art to understand how the researchers are conducting their work on this topic. Section 3 details the protocol used to conduct this works, as well as, the taken assumptions. Section 4 illustrates the results of the experience and performs a detailed analysis of them. Finally, Sect. 5 resumes the work and states some future work to improve the knowledge and technology in his topic.

2 State of the Art

This section reviews the most relevant contributions to this topic in the literature. This review focuses essentially on the deep learning applications for fruit detection.

The lighting conditions in the machine vision is an important issue to consider, in which there are several techniques used to increase the robustness in the detection of fruits, as proposed by Sa et al. [15] analyzes the same images with two different approaches: using RGB color and Near-Infrared (NIR). Its approach included the Deep Convolutional Neural Network (DCNN) architecture with the configuration of the VGG-16 network.

Analyzing in a more specific context in the detection of passion fruit, Tu et al. [18] proposed the use of Multiple Scale Faster Region-based Convolutional Neural Networks (MS-FRCNN) for using RGB color images combined with Depth (RGB-D). The proposed method was able to achieve greater accuracy despite not being the fastest to perform the detection. In a different approach to the detection of cherry tomatoes using regressive methods, Yuan et al. [19] chose to use the Single Shot multi-box Detection architecture (SSD), to compare four different neural networks, among which the Inception V2 was evidenced with an Average Precision (AP) of 98.85%.

The need to perform fruit detection in real-time, Bresilia et al. [6] resorted to the use of the YOLO neural network, with changes in the image input grid and elimination of some layers of the model, to obtain the best relationship between speed and accuracy. Its results were very promising with 95% of the detected fruits.

The approach of Liu et al. [13] based on the YOLOv3 model to create a new model YOLO-Tomato, dedicated specifically for the detection of tomatoes with greater precision. Even with influences caused by the variation of natural light, problems of occlusion, and overlap, they obtained results of approximately 94% of precision in the detection of tomatoes.

In a specific analysis in the apple detection, Briffis et al. [5] chose to use an Adaptative Training Sample Selection (ATSS) DL approach, based on the Resnet 50 and Feature Pyramid Network (FPN) as a backbone. The importance of testing the robustness of detecting fruit under different weather conditions, images with different types of noise and blur were considered for the evaluation. They obtained a maximum value of 94.6% of average precision. The use of deep learning for fruit detection has been widely used and the proof of this is the approach of Zhang et al. [20] which offers the adaptation of a DL architecture to detect different fruits. They proposed a new architecture based on the Multitask Cascaded Convolutional Network (MCCN) called Fruit-MCNN, as well as an augmentation method known as fusion augmentation (FA).

Aiming to detect mango fruit, Koirala et al. [11] compared the performance of six deep learning architectures: Faster R-CNN(VGG), Faster R-CNN(ZF), YOLOv3, YOLOv2, YOLOv2(tiny), and SSD. Also, a new architecture MangoYOLO was developed and trained using different datasets, to create the MangoYOLO models 's', 'pt', and 'bu'. MangoYOLO(pt) achieved an F1 score of 0.968 and Average Precision of 0.983, outperforming the other algorithms, with a detection speed of 8 ms per 512×512 pixel image and 70 ms per image (2 048×2 048 pixels). MangoYOLO(bu) achieved an F1 score of 0.89 on a daytime mango image dataset. This new model was robust when used with images of other orchards, cultivars, and lighting conditions.

Fruit detection in orchards can be quite challenging since there are a number of environment variances. That said, LedNet, a fast implementation framework of a deep-learning-based fruit detector for apple harvesting, was developed by Kang et al. [9]. The model adopts a feature pyramid network and atrous spatial pyramid pooling to improve its detection performance. LedNet achieved 0.821 and 0.853 on recall and accuracy, and its weights size and inference time were 7.4 M and 28 ms, respectively, proving its robustness and efficiency when performing real-time apple detection in orchards.

3 Materials and Methods

3.1 Data Acquisition and Processing

The fruit must be detected, or else it cannot be harvested. This sentence gives the motto for this work, whose main goal is to train and evaluate different Deep Learning (DL) models for tomato detection and classification, supporting the development of automatisms for robotic harvesting in a greenhouse. Since many DL models are characterized as supervised Machine Learning (ML) algorithms, it implies to be trained they must be provided with an annotated dataset. In this work we also seek to compare the performance of an existing public dataset, the Open Image Dataset (OID) [3], with a newly collected dataset, when training and evaluating these models.

Therefore, new images of tomato plants were collected in a greenhouse located in Barroselas, Viana do Castelo, Portugal (Fig. 1b) using a ZED camera[1]. The AgRob v16 robot (Fig. 1a), controlled by a human operator, was guided through the greenhouse inter-rows and captured images of the tomato plants, recording them as a video in a single ROSBag file.

The video recorded by the robot was converted into images by sampling a frame every 3 s, to reduce the correlation between images but ensuring an overlapping ratio of about 60%. This resulted in a dataset composed of 297 images with a resolution of 1280×720 px each.

[1] See https://www.stereolabs.com/zed/.

The images were manually annotated using the open-source annotation tool CVAT [17], considering only the class “tomato”. After being annotated, the images were exported under Pascal VOC format [8], which resumes the annotations of each image in a single XML file.

(a) (b)

Fig. 1. (a): Entrance of the greenhouse where the images were collected. (b): AgRob v16 robot.

High-resolution DL models are time and computationally-consuming. Besides, DL models already available in the state of the art consider the input of square images. Thus, the images had to be resized and split into a resolution of 720×720 px to avoid distortion. The dataset increased to 594 images.

To expand and add variability to the dataset, a process of augmentation was used, which allows various types of transformations that can be applied to an image. The transformations used were applied with a random factor and are as follows: Angle (a); Blur (b); Flip (c); Hue Saturation (d); Multiply (e); Noise (f); Combination1 (g), which applies a transformation randomly; Combination3 (h), which applies a combination of 3 transformations (a random combination of three of transformations with random values), Scale (i) and Translate (j) (Fig. 2). These changes expanded the dataset to 6055 annotated images.

To further train and validate the different models, the dataset was split into a training set and a validation set with a ratio of 3:1 (75% for training and 25% for validation). The training and validation sets contained 4541 and 1513 annotated images respectively.

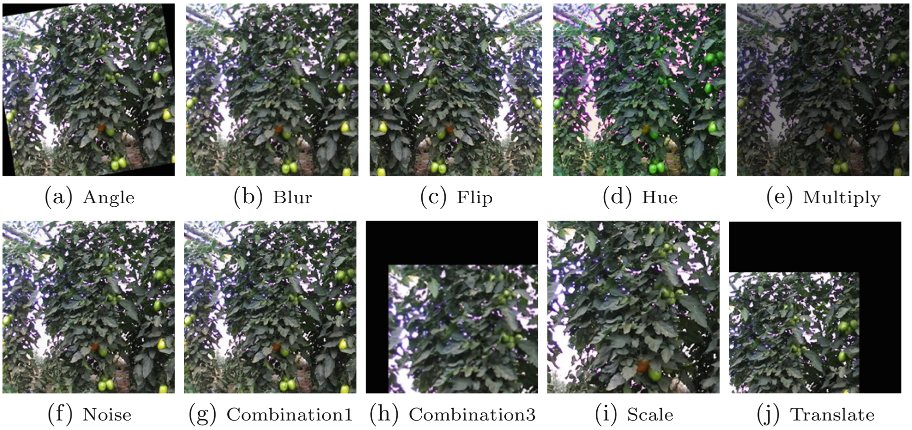


Fig. 2. Different augmentation transformations applied to an image from the dataset.

To infer about the models, we used an independent test set, acquired under the same conditions as the dataset used for train and validation. The data of this set is the same for all the models, is composed of 304 images with a resolution of 720×720 px, as a result of the resized of 1280×720 px images, just as it was done for the other sets.

Regarding the OID v6 public dataset, we choose to use 15 different classes of fruits, including strawberry, tomato, apple orange, grape, lemon, banana, grapefruit, watermelon, pineapple, peach, pear, pomegranate, mango, melon. The training with the OID was proposing to compare a DL model with multiple classes fruits against specific datasets and was not used to training a class non-tomato.

3.2 Training and Evaluating DL Models

Four DL models were trained and evaluated for tomato identification and segmentation, in a greenhouse context, with the OID v6 dataset, and with an acquired dataset, as mentioned earlier.

We considered 3 pre-trained SSD models from the TensorFlow database and 1 pre-trained YOLO model from the Darknet database: SSD MobileNet v2; SSD Inception v2; SSD ResNet 50 and YOLO v4.

All the models were pre-trained with Google's COCO dataset [12]. Both training and inference scripts were run on Google Collaboratory (Colab) notebooks.

Through transfer learning, the pre-trained models were fine-tuned. Slight changes were made to the default training pipeline, most notably adjusting the batch-size for each model (Table 1) and to the optimizer, giving preference to the Adam optimizer for its ease of implementation, low memory requirements, for being computationally efficient, and well suited for problems with a large dataset and/or parameters [10].

Usually, the training sessions ran for 50,000 epochs, a reference value previously established by us. However, for some models this value was not enough to train these models successfully, in some cases, we used one of the training metrics, the "average loss", and stopped training when the curve, from the graph generated by this metric, converged. An evaluation session occurred every 50 epochs.

Table 1. Training batch size for each model.

DL model	Batch size
SSD MobileNet v2	24
SSD Inception v2	32
SSD ResNet 50	8
YOLO v4	64

To evaluate the models, we used the metrics defined by the Pascal VOC challenge [7] (Precision x Recall curve and Mean Average Precision), with the addition of the following metrics: Recall (1), which is the model's ability to detect all relevant objects, Precision (2), the model's ability to identify only relevant objects, and F1 Score (3), the first harmonic mean between Recall and Precision.

$$Recall = \frac{True\ Positives}{All\ groundtruths} \tag{1}$$

$$Precision = \frac{True\ Positives}{All\ detections} \tag{2}$$

$$F_1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

When evaluating the 4 models, we used the OID v6 dataset and an independent dataset collected under the same conditions as the dataset used for training and validation, to identify tomatoes. Regarding the inference process, the Google Colab server was used in all cases with the Tesla T4 GPU and 12 GB VRAM.

4 Results and Discussion

In this section we intend to analyze and evaluate the result of artificial neural networks (ANN) in tomato detection, using two different datasets. The quality and robustness of a dataset in deep learning are essential to achieve the main goal of detecting and classifying an object. To understand its relevance in a real case of tomato detection, it is important to compare the use of a specific dataset against a public dataset with multiple classes. We carried out the model's evaluation as stated in Sect. 3.2.

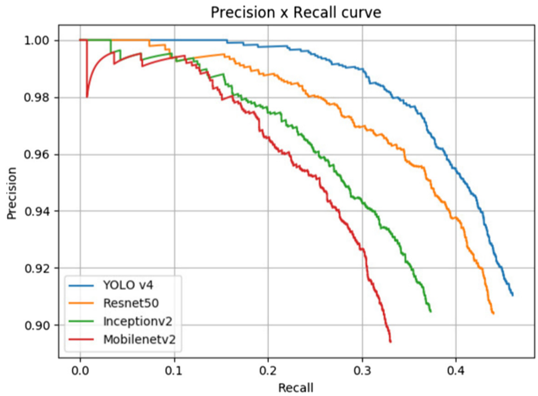
To discuss the different results, the evaluation steps were divided into two phases, the first consists of the analysis of the results of the models trained with the acquired tomato dataset and the second with the same models but trained with OID v6. The main goal is to compare the two different datasets between the four neural networks, evaluating their performance using an inference algorithm. To obtain results, confidence greater than or equal to 30% was considered. The choice of this value is sustained according to the visual perception analysis of the validation data, with the objective of maximize the F1 score. It was also considered, a 50% IOU as default in obtaining the results shown in the Table 2.

Table 2. Results of the different SSD and YOLO models evaluation, considering a 30% predictions and IOU of 50%

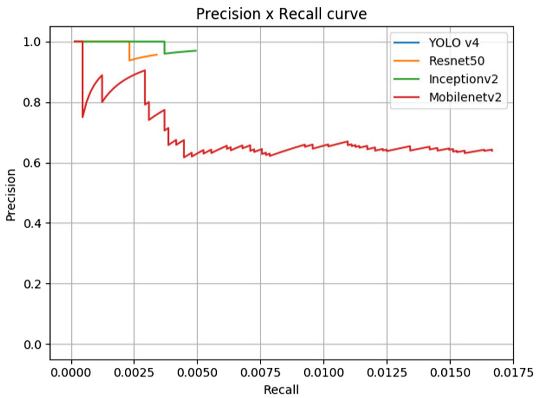
Model	Dataset	mAP	Precision	Recall	F1
YOLO v4	Acquired tomato dataset	45.34%	91.03%	46.05%	61.16%
Resnet 50	Acquired tomato dataset	42.99%	90.39%	44.01%	59.20%
Inception v2	Acquired tomato dataset	36.23%	90.46%	37.32%	52.85%
Mobilenet v2	Acquired tomato dataset	32.07%	89.40%	33.08%	48.29%
YOLO v4	OID v6	0.0%	0.0%	0.0%	0.0%
Resnet 50	OID v6	0.33%	95.65	0.34%	0.68%
Inception v2	OID v6	0.49%	96.97%	0.49%	0.98%
Mobilenet v2	OID v6	1.18%	63.91%	1.67%	3.25%

Regarding our tomato dataset, Table 2 shows good results, namely, the precision obtained with the degree of confidence considered, with YOLOv4 standing out positively followed by Resnet50. Despite these results, the precision x recall ratio was lower than expected, caused essentially by false positives. This relationship can be seen in Fig. 3a. As is to be expected as high-value precision is due to the process of a degree of confidence. Despite the high-level values, the Mobilenet v2 obtained the worst performance followed by Inceptionv2, in which both remained below a 40% recall. Generally analyzing the various models using our tomato dataset, in the Table 2 the high precision values contrast with the low recall values and F1 score. Essentially, these results dues to various noise and false-positive predictions. In a visual comparison, directly between the ground truths and the detections, it is noticed that in conditions of the real environment, problems arise like tomatoes in the background and clustering problems.

In general, our tomato dataset can provide good accuracy vs recall in tomato detection, contrasting with the poor results of OID v6. In Fig. 3b it is possible to conclude that the use of YOLO v4 did not result in any tomato detection. However, Resnet50 can detect false positive fruits. It is important to remember that in the case of our tomato dataset, augmentation was made to increase their robustness, to decrease this typical problem. The purpose of using OID v6 was to understand whether it is a dataset capable of being used not only for validation but also for training deep learning models to detect tomatoes in greenhouses.



(a)

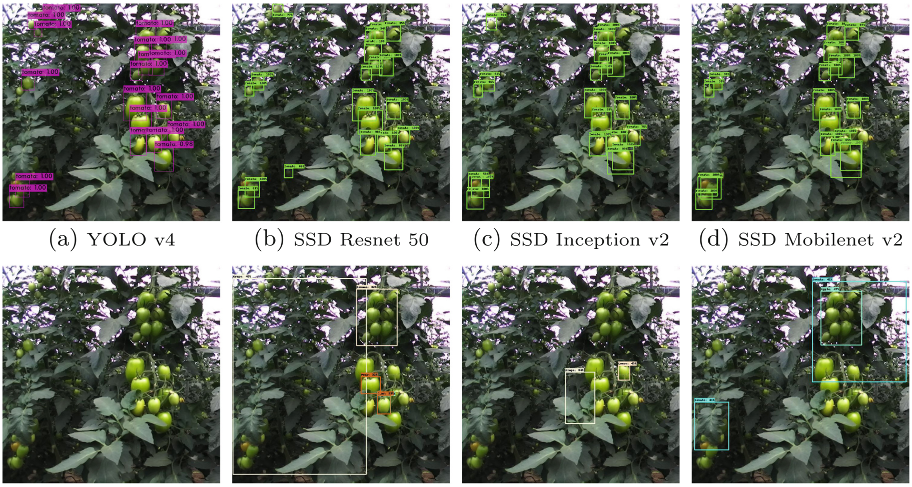


(b)

Fig. 3. Precision x Recall in the test dataset with 30% confidence; (a) curves using our tomato dataset; (b) curves using Open Image Datatset v6;

The use of several classes made the training of the respective models very difficult and time-consuming. The different classes can also be a problem, namely those that contain small datasets and with weak robustness. After inference with the test dataset, it was possible to understand that in the use of OID v6 several incorrect detections were made, namely the labels of bounding boxes with different fruits, among which included in the 15 classes, like grapes, banana, apple, lemon, etc. (Fig. 4). Among the various incorrect detections, we highlight the clustering of tomatoes as a single fruit and detections of leaves as a fruit.

The poor results of OID v6 are directly associated with the tomato class dataset, with images of ripe tomatoes (red) and with few tomatoes per image, contrasting with the images in which it is intended to detect tomatoes. Agricultural companies try to maximize their profits whenever possible and in the case of tomatoes in Portugal, they must be harvested at a relatively early stage of the crop. The reason is due to the speed of maturation of the fruit from its harvest until reaching the final customer.



(a) YOLO v4 (b) SSD Resnet 50 (c) SSD Inception v2 (d) SSD Mobilenet v2

(e) YOLO v4 (f) SSD Resnet 50 (g) SSD Inception v2 (h) SSD Mobilenet v2

Fig. 4. Results comparison with four DL models against two different datasets; (a, b, c, d) DL models with tomato dataset; (e, f, g, h) DL models with OID v6; (Color figure online)

In the analysis of the inference of the test images, it is possible to observe an example (Fig. 4) in which the four neural networks are compared. In the top zone (a, b, c, d) they represent the respective visual results of the models trained with our tomato dataset. It is possible to understand that in general tomato detections using our tomato dataset are very successful, even in cases of occlusion by leaves, variations in lighting, and clustering problems.

In another perspective of analysis, the bottom figures (e, f, g, h) represent the general results collected by the four neural networks trained by OID v6. In the case of YOLO v4, no objects were detected, unlike the other three models, with some objects detected, although they were detected as false positives.

5 Conclusion

Object detection is a crucial task in computer vision. The size and quality of the datasets are an important reason for the continuous improvement of the object detection algorithms, especially for deep learning-based techniques. As mentioned, to train four different neural networks, we compared a dataset of our own with a public and larger dataset with multiple classes, the OID v6 dataset.

Results demonstrate that all four models performed better when faced with the acquired tomato dataset. This could be explained by the divergence between the two datasets that were used. The OIDv6 is very varied in terms of the available classes, however, each of its classes offers different amounts of images, penalizing the training of certain classes. However, the wide variety of classes, many of the images have few annotations, as well as few cases in a real environment, such as in the greenhouse culture itself. Specifically, in the case of the tomato class, the dataset is mostly related to ripe tomatoes (red), contrasting with the color of the tomato that we intend to detect (green and sometimes reddish).

In addition, the set of images comes much closer to a real situation, which a harvesting robot would be exposed to, for example, with objects that are identical in terms of type, color, shape, texture in a similar background. In conclusion, it is important to highlight the proposal of this document to use YOLO v4 when using a dedicated dataset for the detection of tomatoes in the greenhouse, with a precision of 91%.

Some characteristics would still be important to address, which are sometimes limited by the amount of data not being sufficiently large and robust, as well as the analysis of situations in a real environment becomes complex with constant challenges. Therefore, the additional future work focus on:

1. Elaboration of a new dataset with the combination of our tomato dataset and the OID v6, for training evaluation;
2. Evaluate the performance of DL models on Field-programmable gate array (FPGA);
3. Evaluate the performance benchmark of the FPGA against GPU.

Acknowledgments. The research leading to these results has received funding from the European Union's Horizon 2020 - The EU Framework Programme for Research and Innovation 2014–2020, under grant agreement No. 101004085.

References

1. Food and Agriculture Organization of the United Nations (2021). http://www.fao.org/faostat/en/#data/QC
2. Fruit and Vegetables - Your Dietary Essentials: the international year of fruits and vegetables. Food & Agriculture org, S.l. (2021)
3. Open Images V6 (2021). https://storage.googleapis.com/openimages/web/index.html
4. World Health Organization (2021). https://www.who.int
5. Biffi, L.J., et al.: ATSS deep learning-based approach to detect apple fruits. Remote Sens. **13**(1), 54 (2020). https://doi.org/10.3390/rs13010054
6. Bresilla, K., Perulli, G.D., Boini, A., Morandi, B., Corelli Grappadelli, L., Manfrini, L.: Single-shot convolution neural networks for real-time fruit detection within the tree. Front. Plant Sci. **10**, 611 (2019). https://doi.org/10.3389/fpls.2019.00611
7. Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL visual object classes (VOC) challenge. Int. J. Comput. Vis., 34 (2010). https://doi.org/10.1007/s11263-009-0275-4

8. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010). https://doi.org/10.1007/s11263-009-0275-4
9. Kang, H., Chen, C.: Fast implementation of real-time fruit detection in apple orchards using deep learning. Comput. Electr. Agric. **168**, 105108 (2020)
10. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv:1412.6980 [cs] (January 2017)
11. Koirala, A., Walsh, K.B., Wang, Z., McCarthy, C.: Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of 'MangoYOLO'. Precis. Agric. **20**(6), 1107–1135 (2019)
12. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
13. Liu, G., Nouaze, J.C., Touko Mbouembe, P.L., Kim, J.H.: YOLO-Tomato: a robust algorithm for Tomato detection based on YOLOv3. Sensors **20**(7), 2145 (2020). https://doi.org/10.3390/s20072145
14. Magalhães, S.A., et al.: Evaluating the single-shot multibox detector and Yolo deep learning models for the detection of tomatoes in a greenhouse. Sensors **21**(10) (2021). https://doi.org/10.3390/s21103569. https://www.mdpi.com/1424-8220/21/10/3569
15. Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., McCool, C.: DeepFruits: a fruit detection system using deep neural networks. Sensors **16**(8), 1222 (2016). https://doi.org/10.3390/s16081222
16. Saha, S.: A Comprehensive Guide to Convolutional Neural Networks - the ELI5 way (2018)
17. Sekachev, B., Manovich, N., Zhiltsov, M.: opencv/cvat: v1.1.0 (2020). https://doi.org/10.5281/zenodo.4009388. https://zenodo.org/record/4009388#.YHcbXD_OUkl
18. Tu, S., et al.: Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images. Precis. Agric. **21**(5), 1072–1091 (2020). https://doi.org/10.1007/s11119-020-09709-3
19. Yuan, T., et al.: Robust Cherry Tomatoes detection algorithm in greenhouse scene based on SSD. Agriculture **10**(5), 160 (2020). https://doi.org/10.3390/agriculture10050160
20. Zhang, L., Gui, G., Khattak, A.M., Wang, M., Gao, W., Jia, J.: Multi-task cascaded convolutional networks based intelligent fruit detection for designing automated robot. IEEE Access **7**, 56028–56038 (2019). https://doi.org/10.1109/ACCESS.2019.2899940