Pest Detection in Olive Groves Using YOLOv7 and YOLOv8 Models

Adília Alves[1,2,3,4(✉)], José Pereira[3,4], Salik Khanal[6], A. Jorge Morais[2,5], and Vitor Filipe[1,5]

[1] Universidade de Trás-os-Montes e Alto Douro, 5000-801 Vila Real, Portugal
aidcalves@gmail.com
[2] Universidade Aberta, 1269-001 Lisboa, Portugal
[3] Centro de Investigação de Montanha (CIMO), Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Bragança, Portugal
[4] Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança, Campus de Santa Apolónia, 5300-253 Bragança, Portugal
[5] LIAAD - INESC TEC, 4200-465 Porto, Portugal
[6] Center for Precision and Automated Agricultural Systems, Biological Systems Engineering, Washington State University, Prosser, WA, USA

Abstract. Modern agriculture faces important challenges for feeding a fast-growing planet's population in a sustainable way. One of the most important challenges faced by agriculture is the increasing destruction caused by pests to important crops. It is very important to control and manage pests in order to reduce the losses they cause. However, pest detection and monitoring are very resources consuming tasks. The recent development of computer vision-based technology has made it possible to automatize pest detection efficiently.

In Mediterranean olive groves, the olive fly (Bactrocera oleae Rossi) is considered the key-pest of the crop. This paper presents olive fly detection using the lightweight YOLO-based model for versions 7 and 8, respectively, YOLOv7-tiny and YOLOv8n. The proposed object detection models were trained, validated, and tested using two different image datasets collected in various locations of Portugal and Greece. The images are constituted by sticky yellow trap photos and by McPhail trap photos with olive fly exemplars. The performance of the models was evaluated using precision, recall, and mAP.95. The YOLOV7-tiny model best performance is 88.3% of precision, 85% of Recall, 90% of mAP.50, and 53% of mAP.95. The YOLOV8n model best performance is 85% of precision, 85% of Recall, 90% mAP.50, and 55% of mAP.50 YOLO8n model achieved worst results than YOLOv7-tiny for a dataset without negative images (images without olive fly exemplars). Aiming at installing an experimental prototype in the olive grove, the YOLOv8n model was implemented in a Ubuntu Server 23.04 Raspberry PI 3 microcomputer.

The authors are grateful to the Foundation for Science and Technology (FCT, Portugal) for financial support through national funds FCT/MCTES (PIDDAC) to CIMO (UIDB/00690/2020 and UIDP/00690/2020) and SusTEC (LA/P/0007/2020).

A. I. Pereira et al. (Eds.): OL2A 2023, CCIS 1982, pp. 50–62, 2024.
https://doi.org/10.1007/978-3-031-53036-4_4

Keywords: Olives sustainable production · Convolutional Neural Network · Deep Learning · YOLOv7 · YOLOv8

1 Introduction

The olive tree (Olea europaea) is one of the oldest plants and it is widespread in different parts of the world with Mediterranean climate. This crop has great economic, social and cultural impact in the Mediterranean basin, and the olive products (olive oil and table olives) are very appreciated and nutritional balanced. Several pests, pathogens, and nematodes affect olive trees, threatening plant health and production and, consequently, causing annual economic losses. Olive fly (Bactrocera oleae Rossi) is one of the most serious pests that attack olives, being considered a key pest in the producing countries of the Mediterranean region. The olive fly larvae causes important losses that can be, directly, caused by pulp consumption, and indirectly, caused by the decrease in product quality. When attacked, the quality of fruits which are destined to olive oil extraction is affected (e.g., increase of free acidity and peroxide values, decrease of oil stability and organoleptic score). When fruits are destined to table olives, an attack higher than 1–2% makes the production of this product unfeasible.

Nowadays, computer vision-based methods, in particular Deep Learning methods, are revolutionizing agriculture and pest detection pushing toward digital agriculture technology. In recent years, scientists are focusing their research on pest detection and management using computer vision-based models. Ahmad et al. [1], proposed a smartphone-based automated system using an IP-camera to detect insect pests from digital images/videos to reduce farmers' reliance on pesticides. The proposed approach was based on You Only Look Once (YOLO) object detection algorithm including YOLOv5 (n, s, m, l, and x), YOLOv3, YOLO-Lite, and YOLOR. Likewise, Dai et al. [2] proposed an improved YOLOv5m-based model to detect pests in plants. Zu et al. [3] considered three families of detectors to identify and detect pests using Raspberry Pi where the detection of the pest was carried out using faster region-based convolutional neural networks (Faster R-CNN), region-based fully convolutional networks (R-FCN), and single multi-box detectors (SSD). These meta-architectures were combined with deep feature extractors (e.g., VGG nets and ResNet).

Machine Learning/Deep Learning application for automatic pest detection be-came recently, more than an important academic research topic, a broad market application service. Examples of pest detection tools available in the market are Farmesense[1] which counts the prevalence of different insects using its distinctive wing beat sound and RapidAIM[2] which detects different insects using its distinctive behavior (movement and wing beat sound).

The use of electronic and communication devices for monitoring and control of olive plants diseases and pests has made slow progress, mainly in small farms,

[1] https://www.farmsense.io/.
[2] https://rapidaim.io/.

mostly due to devices and system cost, climatic limitations (temperature, humidity, solar exposition) in olive crops, and the absence of good communication means, since internet connection, and even GSM system, can be, and often are, unavailable in olive crops. Even though much research and advances have been made in automatic agricultural pest detection regarding deep learning-based object detection, it is always valuable to consider the robustness, reliability, and cost-effectiveness of the system.

Last years, YOLO family models, thanks to its fastness and accuracy performances, have been used with success to detect insects [4]. So, it is relevant to test more recent YOLO family models (YOLOv7 and YOLOv8) in the context of pest detection. The present study evaluated the two more recent YOLO family object detection methods - YOLOv7 and YOLOv8 for olive fly detection in olive groves. In this article, YOLOv7 and YOLOv8 algorithms are compared in object detection mode (training, validation, and detection) with its lightweight models (optimized for inference on edge devices), using the same training, validation, and test datasets, and running the algorithms in the same hardware and software infrastructure. This study is part of a wider project to create a technically feasible system for the detection/prediction of olive fly pest attacks using small, affordable devices (edge devices) to decrease the cost of the olive fly detection system.

YOLOv7 is an anchor-based single-stage object detection algorithm whereas YOLOv8 is a center-based anchor-free single-stage object detection algorithm. In anchor-based object detection algorithms, the process to locate objects begins with the identification/location of potential bounding boxes (anchors), the selection between these bounding boxes of the most promising ones to match objects (the distinction between positives and negatives), and finally slightly movement and resizing of the selected bounding boxes, as necessary, to obtain the best possible fit of the bounding boxes to the objects. Center-based detectors are like anchor-based detectors, as they treat points (instead of anchor boxes) as preset samples. Center-based detectors begin with just one point (center) per object and use spatial and scale constraints to select samples (the distinction between positives and negatives) looking for the best possible fit of the bounding boxes to the objects.

According to Zhang, Chi, Yao, Lei and Li [5], "The essential difference between anchor-based and anchor-free detection is actually how to define positive and negative training samples, which leads to the performance gap between them. If they adopt the same definition of positive and negative samples during training, there is no obvious difference in the final performance, no matter regressing from a box or a point."

The smallest models (YOLOv7-tiny and YOLOv8n) based on the number of deep learning layers and functionality were used for the experiments to maintain the affordable speed in less powerful computing devices. For example, the YOLOv7 neck network, more precisely, its FPN/PAN feature fusion method adds a top-down pathway and lateral connections to the regular bottom-up pathway (normal flow of a convolution layer) allowing the fusion of received features into three feature maps at multiple scales with semantically strong features. According to Li, Xu, and Zhong [6] YOLOv7 utilizes the combination of FPN and PAN for feature fusion, which cannot guarantee detection accuracy in different resolution images, leading to poor performance in detecting objects with varying shapes and sizes. In present study, the olive fly exemplars have similar body size and well-preserved physiognomy.

This paper is divided into the following sections: The second section introduces the datasets, the deep learning models used to detect olive flies exemplars in traps and the metrics used to compare models performance; The third section focuses on the comparative experiments and the obtained experimental results; The fourth section is the conclusion and the direction of subsequent work and improvement.

2 Materials and Methods

2.1 Dataset Description

All the proposed models were trained, validated, and tested using two custom datasets: a) one dataset constituted by yellow sticky trapped insects images collected in Portugal [7], (Dataset-1) and another dataset constituted by three smaller datasets (Dataset-2). Dataset-2 includes Dataset-1 and two other datasets: a dataset of McPhail-type trapped insects images [8] (Greck dataset), and a dataset of yellow sticky trapped insects images without olive fly exemplars [9] (md-121 dataset). Dataset-1 contains 513 RGB images of olive groves captured by mobile phones of diverse brands (e.g. TCL 10 Plus T782H 6 GB/256 GB, HUAWEI P40 LITE JNY-LX1 6 GB/128 GB) in a vineyard located in Vila Real or Bragança districts of Portugal (Trás-os-Montes). The images were collected between September and November 2022. The image resolution is 72ppi (ppi-points per inch) or 96ppi and the image dimension is 2448×3264, 3840×2160 or 4000×3000. In the images, the olive fly exemplars body measures about 60px and are not overlapped (but sometimes very close to one other). All olive fly exemplars captured by trap are roughly the same size and are presented in a lateral or dorsal position. The dataset images have shadow and tilt areas. Almost all the olive fly exemplars have a yellow background with black lines (the look of the yellow sticky trap) (see Fig. 1)

As previously mentioned, Dataset-2 is constituted by three smaller datasets: Dataset-1, Greek dataset and md-121 dataset.

The Greek dataset consists of 542 images with a pixel per inch of 72 ppi with diverse dimensions (450×2015px to 2064×1161px). The olive fly exemplars body measures around 40px. This dataset, according to Kalamatianos et al. [8], was collected from year 2015 to 2017 in various locations of Corfu, Greece. As the images were collected using different devices (smartphones and tablets running the e-Olive app, photo-cameras available at the field during trap inspection, etc.) the images of the dataset are not standardized in terms of resolution and dimension. In the images, the olive fly exemplars' body measures about 60px, is not overlapped and also has a well-preserved physiognomy. All olive flies'

Fig. 1. The images of olive flies collected where the olive flies are trapped in a yellow background trapper. (Color figure online)

exemplars captured by trap are the same size and are presented in a lateral or dorsal position (see Fig. 2).

Fig. 2. The images of olive flies collected with McPhail Traps

The md-121 dataset consists of 283 images of yellow sticky traps that contain insects but no olive fly exemplars. According to Deserno and Briassouli [9] the small size of objects, occlusions, similarity in appearance and other factors are the major characteristics of the images. The images were taken in laboratorial environment, so they have less visual noise (shadow, tilt). All images have the dimension 5184×3456px and the resolution 72ppi (see Fig. 3).

2.2 Experimental Design

YOLOv7-tiny v0.1 and YOLOv8n version 8.0.59 were trained in a 12th Gen Intel Core i9-1290K 3.2 GHz CPU with 32.0 Gb RAM and with a NVIDIA GeForce

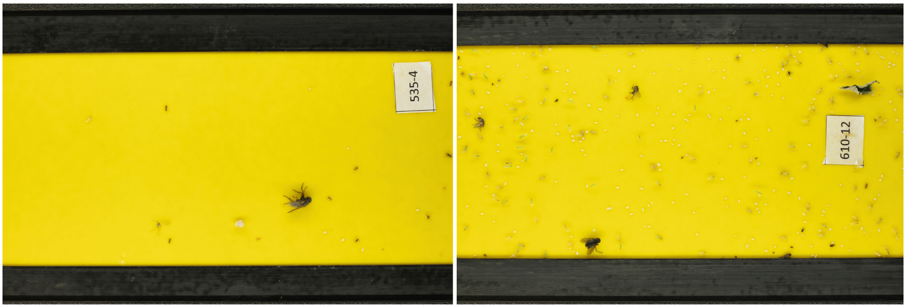

Fig. 3. Dataset3 collected in laboratorial environment.

RTX 4080 16 GB, using Jupyter Notebooks and GPU in Visual Studio Code 1.77.1, running in Windows 10 (with the CONDA version 4.12.0, programming language Python 3.10.9, and the deep learning framework Pytorch 1.13.1).

All the images were labeled using LabelImg tool by an olive-growing researcher producing a text file in YOLO format (x-top left, y-top left, width, height) for each dataset image. Note that all olive fly exemplars (both male and female) were encoded as 'dracus', so just one class object was considered for YOLO detection and all other insects present in the image were ignored (considered background). All the experiments were carried out on Roboflow platform[3] which requires uploading the labeled dataset to platform. For training, it generates several versions of the original dataset using preprocessing and augmentation operations. After splitting the dataset randomly into the train, test, and validation folders, with respectively 80%, 10% and 10% of images, the dataset was exported into YOLOv7 PyTorch TXT and YOLOv8 formats (which includes a config file with YAML extension and the images and annotations distributed by three folders - train, test and validation) and used in YOLO training and detection. Care has been taken to ensure that the three folders, after randomly division of Dataset-2's images, have images from its three constituent datasets.

Both YOLOv7 and YOLOv8 resized input image maintaining proportions and padding the smaller dimension (width or height), by default, to 640×640. As insects are small targets in images and the default initial resize of YOLO model reduces even more the insects size, in this study the input images size is defined as 1280. The input images size is not bigger than 1280px because of the processing and memory limitations of the training system and because of the constraints defined by the device that will capture future images - ESP32-CAM - in the broader system for olive fly's detection in olive groves.

First, several combinations of hyper parameters (e.g. image rotation angle, zoom scale, use or not of mosaic) were used in several trainings and no considerable variation in the results was obtained, meaning that the not so good performance results of YOLOv8n (essentially with Dataset-1) is not in the tuning of these parameters. Thus, the YOLOv8n's worst performance results either

[3] https://roboflow.com/.

from the lack of discriminant signal of the dataset (underfitting) or from the adequacy of the YOLOv8's neuronal structure to the olive fly detection in the traps (over-fitting or underfitting).

2.3 Evaluation Metrics

The performance of the object detection models was evaluated using Precision, Recall, mAP and mAP.95. Precision measures the accuracy in positive prediction.

$$Precision = \frac{True\,positive}{True\,positive + False\,positive}$$

Recall measures the proportion of actual positive cases that the model correctly identifies.

$$Recall = \frac{True\,positive}{True\,positive + False\,negative}$$

There is often a trade-off between precision and recall; for example, increasing the number of True Positives (higher recall) can result in more False Positives too (lower precision). To account for this trade-off, the Average Precision (AP) metric and its average over all categories, the Mean Average Precision (mAP) metric incorporate the precision-recall curve that plots precision against recall for different confidence thresholds.

$$AP = \frac{1}{n}\sum_{i=1}^{n} p(\gamma i),$$

where $p(\gamma i)$ is the Precision at each chosen confidence threshold $\gamma 1, \gamma 2, ...$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} p(AP\,i)$$

where APi is the Average Precision for each class.

In present study just a class will be detected (olive fly), so, AP and mAP will have the same value. The mAP.0.5 is the mean average precision at the intersection over union (IoU) threshold of 0.5. The mAP.95 is the average mAP over different IoU thresholds, ranging from 0.5 to 0.95.

$$IoU = \frac{area(gt \cap pd)}{area(gt \cup pd)}$$

where gt is the ground truth and pd is the prediction.

3 Results and Discussion

YOLOv7-tiny performs better than YOLOv8n for Dataset-1 and both models perform similarly for Dataset-2, as can be seen in Table 1. To analyze and understand weaker experimental performance results obtained by YOLOv8n for Dataset-1 changes on the Dataset-1 and on the YOLOv8n model were made, as follows. The number of Backbone layers was reduced for model YOLOv8n (with remotion of the three final layers) to check if YOLOv8 weaker performance results from overfitting introduced by the large YOLOv8 structure relative to the small object detection. Note that the operation of layers' removal in YOLOv8n model obligates to train the model straight from the beginning (ignoring pre-train with COCO dataset). This YOLOv8n training performance did not improve, on the contrary, it got worse (Precision: 0.652; Recall:0.742; mAP.50: 0.707 and mAP.95: 0.331). A larger version of YOLOv8 was tested - YOLOv8s - and the training performance did improve significantly. Analyzing the two previously mentioned YOLOV8 structural changes implemented, we can conclude that the worst performance of YOLOv8n with Dataset-1 results essentially from features' extraction or from definition of positive and negative samples during training. Some recognized problems with the two used datasets are, mainly with Dataset-1, visual noise (shadow and tilt) and small objects.

The difficulty of YOLO family algorithms in small object recognition is identified and reported by literature [5,10]. Some published workarounds to this problem are: to change the detection model or to change the dataset. Feasible options to change the dataset are: a) increase image resolution; b) increase model input resolution; c) tilt the images; d) generate more data augmentation. Table 1

Table 1. Table with training configurations and resulting evaluation metrics.

Model	Dataset	Size (pixels)	Precision	Recall	mAP.50	mAP.95
YOLOv7-tiny	Dataset-1	1280px	0.817	0.872	0.900	0.529
YOLOv7-tiny	**Dataset-2**	**1280px**	**0.8827**	**0.845**	**0.9038**	**0.531**
YOLOv8n	Dataset-1	1280px	0.738	0.712	0.776	0.378
YOLOv8n	Dataset-1 with bounding box augmentation	1280px	0.62144	0.546	0.547	0.178
YOLOv8n with less background layers	Dataset-1	1280px	0.652	0.742	0.707	0.331
YOLOv8s with COCO[3] pre-train	Dataset-1	1280px	0.725	0.727	0.774	0.393
YOLOv8n	**Dataset-2**	**1280px**	**0.854**	**0.854**	**0.905**	**0.55**

[3] The MS COCO (Microsoft Common Objects in Context) dataset is a large-scale object detection, segmentation, key-point detection, and captioning dataset. The dataset consists of 328K images

Results with Higher Resolution Images

This study is part of a wider project to create a technically feasible system for the detection/prediction of olive fly pest's attack using small, affordable devices

- edge devices - to collect and process traps' photos. The tested edge device used to capture photos' traps is the ESP32-CAM module (with video camera OV2640). The maximum resolution captured by this device is 1600×1200px.

The same models were tested with two datasets (Dataset-1 and Dataset-2) at 1280×1280 input resolution and gave the better accuracy for Dataset-2 (a larger and more heterogeneous dataset). Increase YOLO input resolution increases results, however, YOLO input resolution cannot be greater than image input resolution, so, the model's input resolution must be 1280×1280px.

This study augmented the number of images dataset but YOLOv8n model's training results did not be significantly improved. Filtering out extraneous class is not possible for present study because the only labelled class in images is the olive fly. All other insects are considered background because they are not labelled. The Portuguese images' background (outside the trap) was removed, trying to remove false positives. The training results did not improve, so it seems that YOLOv8 features' extraction problem is only in discriminating what is an olive fly exemplar and what is another insect exemplar. In order to increase detection accuracy of images with visual noise (for example, shadow areas and tilt), some authors find ways to automatically detect and remove such visual problems [11,12].

In order to increase detection accuracy of small objects in YOLO family algorithms without loosing the detection accuracy of other objects, some changes to detection models are suggested in literature: improve the Backbone module in order to make the context information saved in the feature extraction process more complete; improve feature fusion method of different scaled feature maps; improve identification/location of potential objects [4,10]. Finally, the inference/detection was performed, using a minimal confidence of 0.25 and the weights (best.pt) obtained in the training phase by 'YOLOv7-tiny with COCO pre-train' for Dataset-2 for three different datasets, described bellow.

A dataset constituted by 11 trap photos captured in the field. Some examples of the resulting inference executed with images of this dataset are presented bellow (see Fig. 4).

A dataset constituted by 12 trap photos captured in the laboratory environment. These photos have exclusively olive fly exemplars. Some examples of the resulting inference executed with images of this dataset are presented bellow Fig. 5. A dataset constituted by 41 trap photos captured in laboratory environment with-out olive fly exemplars. Some examples of the resulting interference executed with images of this dataset are presented bellow (see Fig. 6).

The inference executed in previously described datasets achieved very good results. The probability boxes are positioned around the detected objects with associated confidence, and as can be seen, in Figs. 4, 5 and 6, most of the pests were correctly identified with good confidence. As can be seen in Fig. 6, in a dataset without olive flies no probability box was placed, so, no False Positive was detected.

Fig. 4. Result of inference executed for dataset captures in the field (with olive fly exemplars and other insects)

Fig. 5. Result of inference executed for dataset captured in laboratory (with olive fly exemplars).

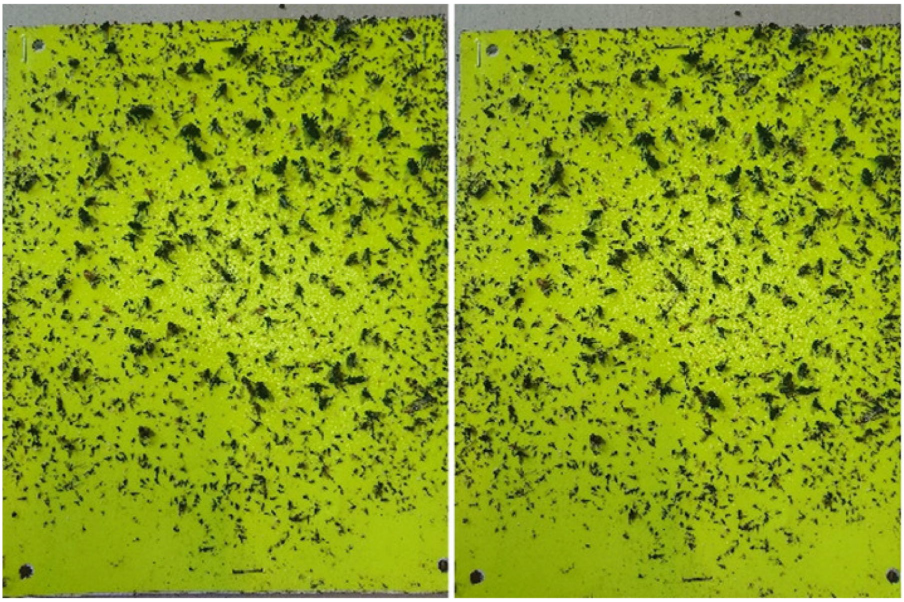

Fig. 6. Result of inference executed for dataset captured in laboratory (without olive fly exemplars).

4 Conclusions and Future Work

YOLOv7-tiny and YOLOv8n models obtained better results for Dataset-2 than for Dataset-1. YOLO8n model achieved slightly better training performance results than YOLOv7-tiny for Dataset-2 and much worst results than YOLOv7-tiny for Dataset-1. Dataset-2 introduced, relatively to Dataset-1, false training samples using md-121 dataset to reduce the false alarm rate (False positives number). YOLOv7-tiny Image segmentation can be perceived as a further extension of object detection because, instead of enclosing each object in the image with a bounding box, it marks each object's pixel through pixel-wise masks. Image segmentation is not necessary for pests detection in photos because the shape of the pest brings no relevant information for the pests detection task. Although YOLO family models give good results for medium to large objects, detection of small objects in messy images performs not so well. As future work: a) improvements must be done to the dataset images (for example with noise remotion and olive fly body preservation) and to the object detection model in order to improve olive fly detection. All traps' photos will be captured in the field, so, some important and urgent steps are: to fix the trap on a fixed structure in such a way as to prevent the trap borders from bending; and use yellow sticky traps that have not printed logotypes and black lines. This will drastically reduce images noise. b) the Raspberry 3 device must receive photos acquired by ESP32 web CAM devices scattered in olive grove using LoRa (Long Range) technology, count the number of detected olive fly exemplars and communicate the counting using a GSM/GPRS module. The images must be sent once a week and the traps must be cleaned once a month.

References

1. Ahmad, I., Yang, Y., Yue, Y., Ye, C., Hassan, M., Cheng, X., Yunzhi, W., Zhang, Y.: Deep learning based detector YOLOv5 for identifying insect pests. Appl. Sci. **12**(19), 10167 (2022)
2. Min Dai, Md., Dorjoy, M.H., Miao, H., Zhang, S.: A new pest detection method based on improved YOLOv5m. Insects **14**(1), 54 (2023)
3. Zhu, D., et al.: Knowledge graph and deep learning based pest detection and identification system for fruit quality. Internet Things **21**, 100649 (2023)
4. Nithin Kumar, N., Flammini, F.: YOLO-based light-weight deep learning models for insect detection system with field adaption. Agriculture **13**(3), 741 (2023)
5. Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, June 2020
6. Li, B., Chen, Y., Xu, H., Zhong, F.: Fast vehicle detection algorithm based on lightweight yolo7-tiny (2023)
7. Pereira, J.A.: Yellow sticky traps dataset olive fly (Bactrocera Oleae) (2023)
8. Kalamatianos, R., Karydis, I., Doukakis, D., Avlonitis, M.: DIRT: the dacus image recognition toolkit. J. Imaging **4**(11), 129 (2018)

9. (Ard) Nieuwenhuizen, A.T., et al.: Raw data from yellow sticky traps with insects for training of deep learning convolutional neural network for object detection (2019)
10. Lou, H., et al.: DC-YOLOv8: small size object detection algorithm based on camera sensor, April 2023
11. Tatar, N., Saadatseresht, M., Arefi, H., Hadavand, A.: A new object-based framework to detect shodows in high-resolution satellite imagery over urban areas. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XL-1/W5:713–717, December 2015
12. Belde, S.: Noise removal in images using deep learning models, April 2021