USE OF YOLOV5 OBJECT DETECTION ALGORITHMS FOR INSECT DETECTION

Lino Oliveira[1], Margarida Victoriano[1], Adilia Alves[2] and José Pereira[2]
[1]*INESC TEC, Campus da FEUP, R. Dr. Roberto Frias, 4200-465 Porto, Portugal*
[2]*CIMO – IPB, Campus de Santa Apolónia, 5300-253 Bragança, Portugal*

ABSTRACT

Climate change affects global temperature and precipitation patterns that influence the intensity and, in some cases, the frequency of extreme environmental events, such as forest fires, hurricanes and storms. These events can be particularly conducive to the increase of plant pests and diseases, which causes significant production losses. So, the early detection of pests is of the main importance to reduce pest losses and implement more safe control management strategies protecting the crop, human health, and the environment (e.g., precision in the pesticide application). Nowadays, pests' detection and prediction are mainly based on counting insects on attacked organs or in traps by experts, but this is a costly and time-consuming task for extensive and geographically dispersed olive groves. Machine learning algorithms, using image analysis, can be used for autonomous pests' detection and counting. In the present practical work, YOLOv5 was chosen to detect and count the olive fly adults (*Bactrocera oleae* Rossi), a key pest of olives. YOLOv5s architecture of YOLO's algorithm was used to test its efficiency in olive fly detection on a mobile deployment solution. The results obtained were quite satisfactory, and the experimental results obtained have been analyzed and presented, encompassing a set of metrics such as precision, recall, and the mean average precision. This study will be extended for other pests and disease detection in future work. Also, this solution will be integrated into a web-based information and management service (with sensors and e-traps) that remotely detect the presence and severity of pest attacks.

KEYWORDS

Object Detection, YOLOv5, Machine Learning, Sustainable Agriculture, CIMO-IPB Dataset

1. INTRODUCTION

Climate change affects global temperature and precipitation patterns. These conditions, in turn, influence the intensity and, in some cases, the frequency of extreme environmental events. In general, these events can be particularly conducive to the increase of plant pests and diseases. The attack of pests and diseases is one of the main causes of crop losses, causing an average 30% decrease in potential crop production. And, in severe cases, it can destroy all the crops. Agricultural ecosystems are known to be complex, multivariable, and unpredictable. To protect crops, it is important to anticipate the attack of pests and diseases to improve its control by more ecological and economical methods (e.g., precision in pesticide application). In this frame, it is urgent to look for real-time detection methods and prediction models for pests. In the olive crop, the olive fly, *Bactrocera oleae* Rossi, is a key pest of olives, causing significant losses in quantity and quality. Quantity losses result from the direct pulp consumption by the olive fly larvae, and quality losses result from the failure of positive attributes for olive oil and table olives.

Identification and localization of objects in photos or video frame's is a computer vision task called 'object detection'. Several algorithms have emerged in the past few years to tackle this task. Nowadays, one of the most popular algorithms for real-time object detection is YOLO (You Only Look Once) because of its speed and accuracy. YOLOv5 (Glenn Jocher, 2022) was used for olive fly detection and counting, from image analysis, in olive grows - essential data to predict future pests' occurrence and severity. For YOLO's algorithm train, a yellow sticky trapped insects' dataset and a McPhail trapped insects' dataset were used previously and manually tagged with labelImg (Tzutalin, 2015). The unique tagged class identified on the dataset was adults of the 'olive fly', one of the most dangerous olive pests in the Mediterranean basin.

2. STATE OF THE ART

Object detection aims to recognize one or more objects in media, drawing bounding boxes around these objects. There are two main methods for object detection, it can be either a neural network-based or non-neural network approach. For non-neural approaches, it is necessary to identify features using a certain method (e.g., scale-invariant feature transform) and then to do classification using another method (e.g., support vector machine). On the other hand, neural network-based techniques usually do not need to specify features and are able to perform end-to-end object detection. This approach is typically based on convolutional neural networks. Examples of that approach are YOLO and R-CNN. YOLO family was first described by Joseph Redmond (Redmond et al, 2015), and it analyses the whole image during training and testing, so it considers contextual information about classes and object's appearance. After inputting the images into the convolutional network, YOLO completes the prediction of the object's classification and location information according to the calculation of the loss function. YOLOv5 is one of the successors of YOLO that aims to detect objects with high performance and is a PyTorch based version (Nelson & Solawetz, 2020), that supports mobile and web deployment.

2.1 YOLO Architecture

The YOLO model was the first object detector to associate the bounding box prediction procedure with class labels in an end-to-end differentiable network. The network is divided in the input, the backbone, the neck, and the head (Solawetz, 2020). The input terminal is responsible for preprocessing the incoming data, comprising the mosaic data augmentation and adaptative image filling. After inputting the images, the algorithm finishes the prediction of the classification and object location, according to the calculation of the loss function, thus transforming this into a regression problem. The adaptative anchor frame computation that YOLOv5 includes, allows it to automatically set the original anchor frame size whenever the dataset changes, enabling it to adjust to different datasets (Li, et al., 2022). The backbone network is used for pre-training and mainly uses a cross-stage partial network (CSP) to reduce the number of calculations and increase the speed of inference. It also uses spatial pyramid pooling (SPP) to extract feature maps from the input image by splitting the significant features, which helps improve detection accuracy. The backbone can be executed either on GPU or CPU platforms. The feature pyramid structures of FPN and PAN are used in the neck network. PAN aggregates parameters from different backbone levels (Liu, 2018). The neck is composed of a set of layers to blend and combine image features and send them forward to prediction. The head receives the features from the neck and moves on to the bounding box and class prediction phase. The YOLOv5 consists of four models with several architectures. The main difference between them lies in the number of feature extraction modules and convolution kernels at specific locations on the network, so the training time and obtained results will be different. The YOLOv5s model was chosen since it was the most adequate to use, because, in the next steps, the model will be exported to a web/mobile application. The training task is essential to the success of an object detection system. Two techniques are available of special relevance during the training phase: data augmentation and loss calculations. Data augmentation aims to perform transformations to the base training data to expose the model to a larger set of semantic variations relative to the original training set and the images are sent through a data loader to be augmented. On the other hand, it also performs a total loss calculation from constituent loss functions to maximize the mean average precision. The detection task is accomplished by dividing an image into a grid system, and each grid detects objects contained. To predict bounding boxes, the model learns the anchor boxes based on the distribution of bounding boxes in the given dataset, with K-means and genetic learning algorithms.

3. PROBLEM DEFINITION AND PROPOSED SOLUTION

The attack of pests and diseases is the main reason for crop losses. Therefore, it is crucial to anticipate the occurrence of crop pests and diseases to improve and implement control strategies. Detecting pests in real time and applying prediction models are important tools to achieve this. In this context, the proposed technological solution implemented a computer vision module that aims to identify olive flies (the most dangerous fruit pests of olives the Mediterranean basin), allowing to increase the sustainability of the olive

operation. The model was trained using YOLOv5 algorithm with a yellow sticky trapped insects' dataset (images collected on Portuguese olive grove) (Pereira, J. A., 2022) together with a McPhail trapped insects' dataset (images collected on Greek olive groves) (Kalamatianos R., 2018).

3.1 Specifying the Dataset

Initially, images referring to the pest under study were collected. Then the image identification task for training was performed using the labelImg tool, where the objects present in the images were properly identified with a bounding box and tagged. The dataset collected was composed of 341 images, collected in various olive groves locations located in Trás-os-Montes, Portugal, from 2021 September to 2021 October. A Greek dataset, consisting of 848 labelled images collected from 2015 to 2017 in Corfu (Greece), was also used. To reduce false positives, the original dataset was augmented with background images (yellow sticky traps without olive fly specimens) acquired from Portuguese traps (45 images) and from a public dataset (164 images) (Tellez, 2019). The dataset used for YOLO's training was obtained by merging these datasets. After this task, it was necessary to standardize the provided dataset since it was obtained by merging different sources, and it's expectable to have some inconsistencies. Having a consistent and clean dataset is essential, as this is necessary for the success of the detection task. Note that just one class of objects was identified. The annotation files provided were on the pascal VOC format but since the algorithm only accepts the YOLO format, it was developed a Python3 script to handle the conversion between these two formats. In YOLO format, each image has a correspondent text file which contains one line per object. Each line starts with the class number, indexed to zero, followed by the coordinates and width and height of the object's bounding box.

3.2 YOLOv5 Training

The training category contains 80% of the dataset images, and both the validation and test categories correspond to 10% of the dataset images. After the physical organization of the input images and labels, the dataset.yaml file was created to specify the dataset settings: the dataset root directory, the relative paths to the training, validation and test folders, the number of classes to be detected and, finally, their names. After, collecting and standardizing the dataset and building the dataset.yaml file, it is possible to start training. Training was started using the smallest model available (YOLOv5s), all layers, and pre-trained weights. Although the pre-trained weights were used, the model was fine-tuned to the loaded dataset. So, the training process accepts as inputs the dataset.yaml file and the corresponding weights. The image size was chosen to be 640 pixels, as stated previously. The batch size corresponds to a hyperparameter that defines the number of samples to process before updating the internal model parameters. This value can be defined according to the available GPU memory. On the other hand, an epoch is a hyperparameter that indicates the number of times that the algorithm will parse through the entire training dataset. Datasets are usually grouped into batches, and an epoch consists of one or more batches.

4. CASE STUDY AND EXPERIMENTAL RESULTS

The implementation targets of this practical work are the autonomous detection and count of olive flies in olive groves. Images were collected and annotated in the olive groves of Trás-os-Montes, Portugal, and used in conjunction with the Greek dataset to train a network that detects a pest from images, to be applied in images collected there. After training, the YOLO algorithm produced the following graphs, regarding losses and some metrics. The YOLOv5 loss function is composed of *box_loss* that refers to the bounding box regression loss, obtained by the mean squared error, the *obj_loss* that refers to the confidence of object presence is the objectness loss, obtained by the binary cross entropy and *cls_loss* that that is the classification loss, obtained by the cross entropy. Since YOLOv5 algorithm is being used for the detection of a single class, there are no class misidentifications, and the classification error is always zero, so the graphs regarding the train and validation class loss are constantly represented by zero. The precision metric measures how much of the bounding box predictions are correct and the recall metric measures how much of the true bounding box predictions are actually correct. The mAP_0.5 is the mean average precision at the intersection over

union (IoU) threshold of 0.5. The mAP_0.5:0.95 is the average mAP over different IoU thresholds, ranging from 0.5 to 0.95. The following image shows the obtained results in training and validation steps, and it describes the metrics presented above.

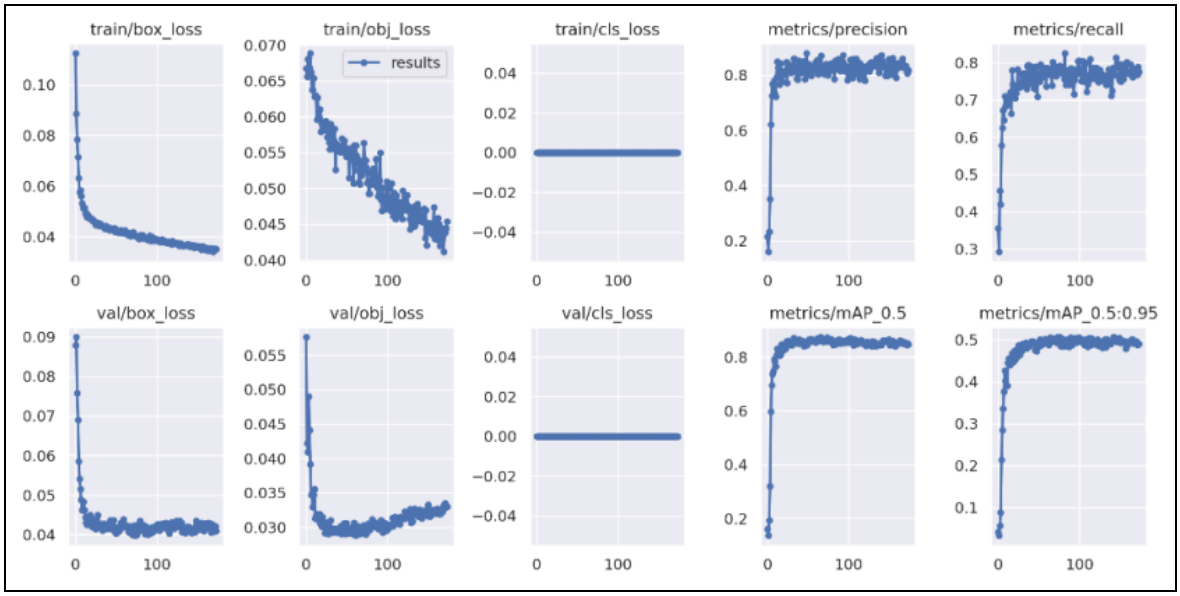

Figure 1. Experimental results

Finally, the inference was performed on a set of four unannotated images and 44 background images from the captures on Portuguese olive groves, using the weights obtained in the training phase.

This process resulted and showed a good performance and excellent detection results. The boxes are around the detected objects with associated confidence, and, as we can see, in the four images below, most of the pests were correctly identified with good confidence.

Figure 2. Inference results

5. CONCLUSION AND FUTURE WORK

This work presents the application of YOLOv5, more precisely of its YOLOv5s architecture, to autonomous olive fly, the most important pest of olives. For YOLOv5 algorithm training, it was used a dataset of trapped insects collected both in Portuguese and Greek olive groves. YOLOv5 algorithm has shown good results and performance. The training of the algorithm will be extended for different pests and disease detection by expanding the input dataset. YOLOv5 training is being improved with dataset's image preprocessing and

with hyper-parameters configuration (e.g., using weighted boxes fusion). In the next phase, robotic traps with sensors will be developed and placed in Trás-os-Montes region, which will automatically collect images of pests in the field. For future work, a web-based information and management system will be developed, integrating this computer vision module, which will receive the images collected from the field by a robotic smart trap, and return the image with the detection results. Since the algorithm was developed in PyTorch, it will be easier to export to mobile and web platforms.

ACKNOWLEDGEMENT

This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project LA/P/0063/2020.

REFERENCES

Glenn Jocher, e. a., 2022. *ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference.* [Online] Available at: https://zenodo.org/record/6222936 [Accessed 1 May 2022].

Kalamatianos R., K. I. D. D. a. A. M., 2018. DIRT: The Dacus Image Recognition Toolkit.

Liu, S. a. Q. L. a. Q. H. a. S. J. a. J. J., 2018. *Path Aggregation Network for Instance Segmentation.* [Online] Available at: https://arxiv.org/abs/1803.01534 [Accessed 5 July 2022].

Li, Z., Tian, X., Liu, X. & Liu, 2022. A Two-Stage Industrial Defect Detection Framework Based on Improved-YOLOv5 and Optimized-Inception-ResnetV2 Models. *Applied Science*, 12(2), p. 834.

Nelson, J. & Solawetz, J., 2020. *YOLOv5 is Here: State-of-the-Art Object Detection at 140 FPS.* [Online] Available at: https://blog.roboflow.com/yolov5-is-here/ [Accessed 18 May 2022].

Redmond, J. e. a., 2015. You Only Look Once: Unified, Real-Time Object Detection. *arXiv.*

Solawetz, J., 2020. *Roboflow.* [Online] Available at: https://blog.roboflow.com/yolov5-improvements-and-evaluation/ [Accessed 18 May 2022].

Tellez, A. (. N. a. J. (. H. a. D. (. J. a. H. (. S. a. L. (. B. a. V. (. S. a. N. (. B. a. E. (. R. a. M. (. d. M., 2019. *Raw data from Yellow Sticky Traps with insects for training of deep learning Convolutional Neural Network for object detection.* [Online] Available at: https://data.4tu.nl/articles/dataset/Raw_data_from_Yellow_Sticky_Traps_with_insects_for_training_of_deep_learning_Convolutional_Neural_Network_for_object_detection/12707066 [Accessed 11 July 2022].

Tzutalin, 2015. *LabelImg Git code.* [Online] Available at: https://github.com/tzutalin/labelImg [Accessed 16 May 2022].