Modelo ya entrenado, listo para usar:

<https://www.tensorflow.org/hub/tutorials/tf2_object_detection>

Basically there are three important drivers to analyze when selecting a model for object detection:

* Accuracy
* Speed of implementation (FPS)

As the user probably will want an immediate response, the speed of implementation has a lot of weight when selecting the model but the accuracy must reach a minimun required level as it is of no use a model that can not detect well.

Regarding the speed of the model, this will depend on which type of Region or single shot based detectors

YOLOv5 >> seems to be a fast training model, with fast implementation/inference so it can be used real time.

official site:

<https://pytorch.org/hub/ultralytics_yolov5/>

tutorial to load the model:

<https://github.com/ultralytics/yolov5/issues/36>

tutorial on how to train for custom dataset/using the pretrained model (training from scratch is also a possibility)

Relevants links to see in more detail:

| Link | Description |
| --- | --- |
| **provided by memorable**  <https://towardsdatascience.com/coco-data-format-for-object-detection-a4c5eaf518c5> | how to use the coco format for object detection |
| **provided by memorable**  <https://github.com/cocodataset/cocoapi/tree/master/PythonAPI/pycocotools> | cocoapi for dealing with the data and its format |
| <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00434-w> | Comparative analysis of deep learning image detection algorithms |
| <https://jonathan-hui.medium.com/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359> | Object detection: speed and accuracy comparison (Faster R-CNN, R-FCN, SSD, FPN, RetinaNet and YOLOv3) |
| <https://amadoruavs.medium.com/comparing-object-detection-algorithms-demystified-cf26c9eb23ef> | Comparing Object Detection Algorithms, Demystified |
| <https://towardsdatascience.com/the-practical-guide-for-object-detection-with-yolov5-algorithm-74c04aac4843> | The practical guide for Object Detection with YOLOv5 algorithm |
| <https://github.com/bhimar/GrocerEye> | GrocerEye - A YOLO Model for Grocery Object Detection |
| <https://blog.roboflow.com/retail-store-item-detection-using-yolov5/> | Retail Store Item Detection using YOLOv5 |
| <https://models.roboflow.com/object-detection> | Summary of a few object detection algorithms |
| <https://blog.roboflow.com/how-to-train-yolov5-on-a-custom-dataset/> | How to Train YOLOv5 On a Custom Dataset |
| <https://tarak-gopani.medium.com/grocery-item-detection-using-tensorflow-object-detection-api-1581fb5df6d6> | Grocery Item Detection using **TensorFlow** Object Detection API |
| <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e> | R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms  Understanding object detection algorithms |
| <https://levelup.gitconnected.com/product-detection-from-grocery-shelf-9db031e0ddc1> | Use Computer Vision for Product Detection on a Grocery Shelf |
| <https://towardsdatascience.com/building-a-toy-detector-with-tensorflow-object-detection-api-63c0fdf2ac95> | Building a Toy Detector with Tensorflow Object Detection API |
| <https://towardsdatascience.com/using-object-detection-for-a-smarter-retail-checkout-experience-3f39acef857b> | Using Object detection for a Smarter Retail Checkout Experience |
| <https://pytorch.org/hub/ultralytics_yolov5/> | **yolov5 official pytorch site** |
| <https://github.com/ultralytics/yolov5/issues/36> | **tutorial on how to load the model** |
| <https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data> | **tutorial on how to train the model on custom data** |