How to train YOLOv3 using Darknet on Colab notebook

Để train được cho Yolo nhận diện các đối tượng đặc thù theo yêu cầu, cúng ta sẽ cần làm các bước lớn sau.

1. Tải source code Darknet – Yolo về máy tính, chỉnh tham số và tiến hành biên dịch (make) source code đó ra file thực thi tùy theo hệ điều hành (window thì là exe, macos với linux thì file bash thì phải, tóm lại là file chạy được)
2. Chuẩn bị dữ liệu train: Hình ảnh của đối tượng bạn định train. Ví dụ như bài này là súng ngắn.
3. Gán nhãn cho dữ liệu: Cụ thể là với từng ảnh trong dữ liệu, chúng ta sẽ gán nhãn cho máy biết đâu là đối tượng cần nhận dạng bằng cách vẽ một hình chữ nhật xung quanh đối tượng đó. Cái này có tool nhé.
4. Tạo các file cần thiết để phục vụ quá trình train, chỉnh sửa tham số train trong file cấu hình Yolo.
5. Chạy lệnh train và ngồi uống cafe đợi.
6. Tận hưởng thành quả bằng cách detect thử một ảnh sample.

**A. Cài đặt Darknet on Colab và test thử kết quả**

Theo hướng dẫn sau:

**https://colab.research.google.com/drive/1WsyEMeIgl02sWei0KmDtnfJywkYuERqQ**

**B. Cách train Yolo để detect các object đặc thù trên Colab**

Bước 1. Chuẩn bị dữ liệu train

Bước 2. Gán nhãn cho dữ liệu

Có 02 công cụ:

*Nếu sử dụng công cụ* ***Bbox-Label-Tool****:*

*Chú ý chọn để cho format là Yolo chứ ko phải là Pascal VOC nhé.*

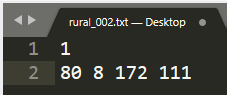
The format of annotations generated by [BBox-Label-Tool](https://github.com/puzzledqs/BBox-Label-Tool) is:

class\_number

box1\_x1 box1\_y1 box1\_width box1\_height

box2\_x1 box2\_y1 box2\_width box2\_height

ví dụ:

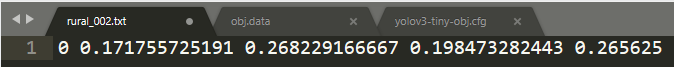


After conversion, the format of annotations converted by [scripts/convert.py](https://github.com/Guanghan/darknet/blob/master/scripts/convert.py) is:

class\_number box1\_x1\_ratio box1\_y1\_ratio box1\_width\_ratio box1\_height\_ratio

class\_number box2\_x1\_ratio box2\_y1\_ratio box2\_width\_ratio box2\_height\_ratio

ví dụ:

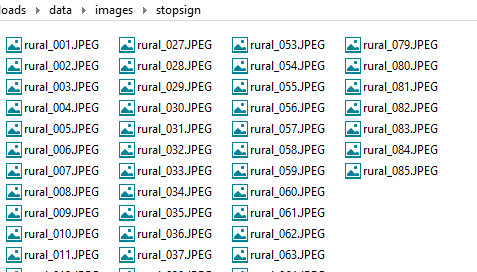


Sau khi thực hiện gán nhãn chúng ta có kết quả sau:

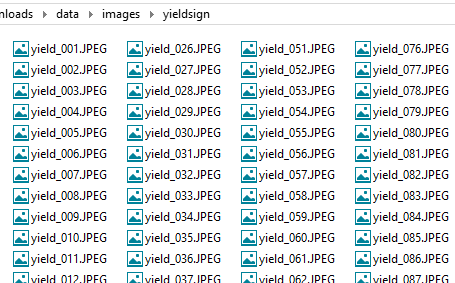
Trong thư mục images:



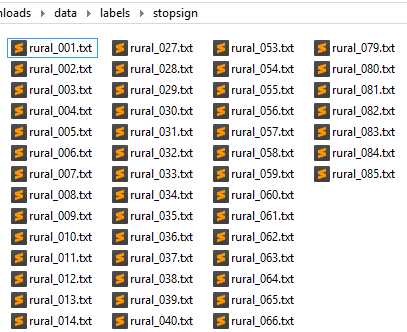
Trong thư mục stopsign



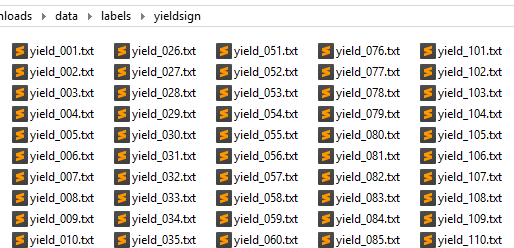
Trong thư mục yieldsign



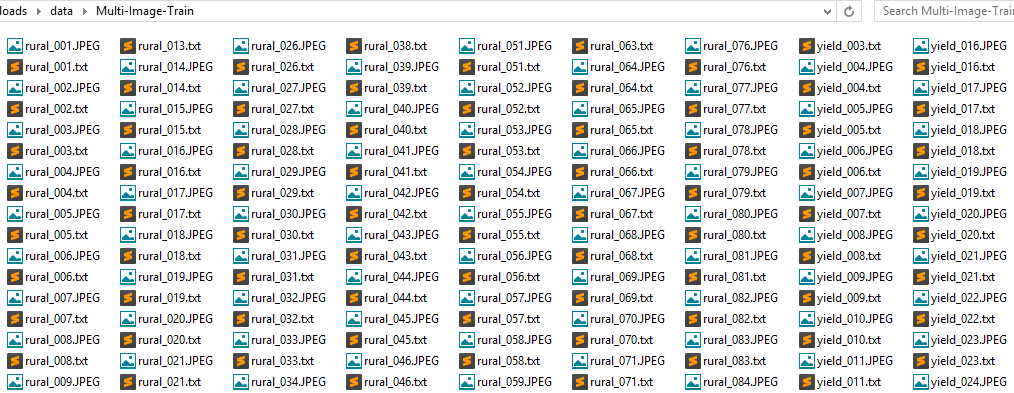
Tương ứng trong thư mục labels cũng có 02 thư mục như thư mục images:



Thư mục yieldsign



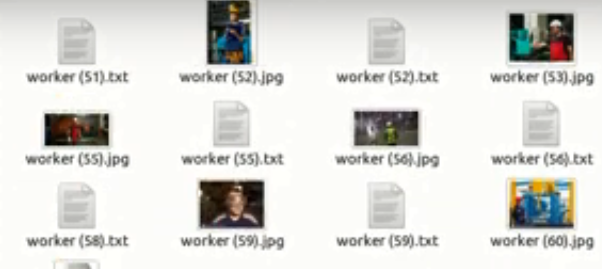
Để chuẩn bị dữ liệu (tách ra tập train.txt và test.txt) chúng ta cần tạo một thư mục chép tất cả hình ảnh, label vào:



*Nếu sử dụng công cụ* ***LabelImg****:*

*Đối với công cụ LabelImg (chúng ta không cần mất thời gian như trên nhé). Chép tất cả hình ảnh vào một thư mục(dù bạn có nhiều nhãn: stopsign, yieldsign,…).*

*Khi sử dụng công cụ này sẽ tạo ra thư mục như hình trên (kaka…)*

**

Hướng dẫn chi tiết: <https://github.com/tzutalin/labelImg>.

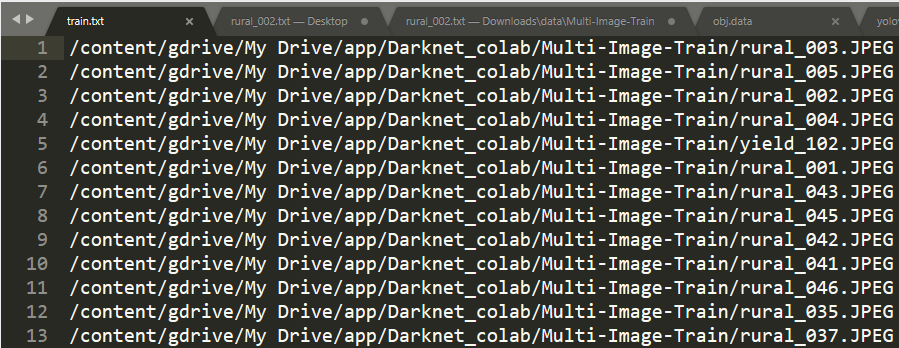
**Bước 4**: Tao file train.txt và test.txt

Khi chúng ta có thư mục như trên, tiến hành phân chia thành 02 tập train.txt và test.txt để chuẩn bị huấn luyện.

Đây là code để tach ra 02 file được chạy trên colab:



Ví dụ nội dung file train.txt



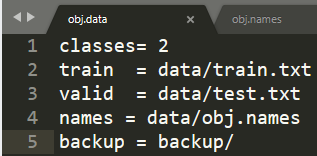
**Bước 5.** Chuẩn bị các file cần thiết phục vụ quá trình train dữ liệu

Chú ý: mục đích sử dụng **tiny-yolo** (to detect your custom objects)

Copy file **train.txt** đã tạo ở trên in the directory **darknet/data**

Copy file **test.txt** đã tạo ở trên in the directory **darknet/data**

Create file **obj.data** in the directory **darknet/data**



Trong đó:

classses = 2 # Số lượng class, ở đây chỉ có 2 đối tượng lên classes=2

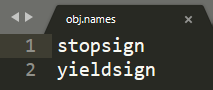
train = data/train.txt # trỏ đến file train của ta thôi

test = data/test.txt # trỏ đến file test của ta

names = data/obj.names # trỏ đến file names làm bên trên

backup = backup/ # là đường dẫn sẽ lưu các file weights trong quá trình train

Create file **obj.names** in the directory **darknet/data**



**yolov3-tiny.weights**

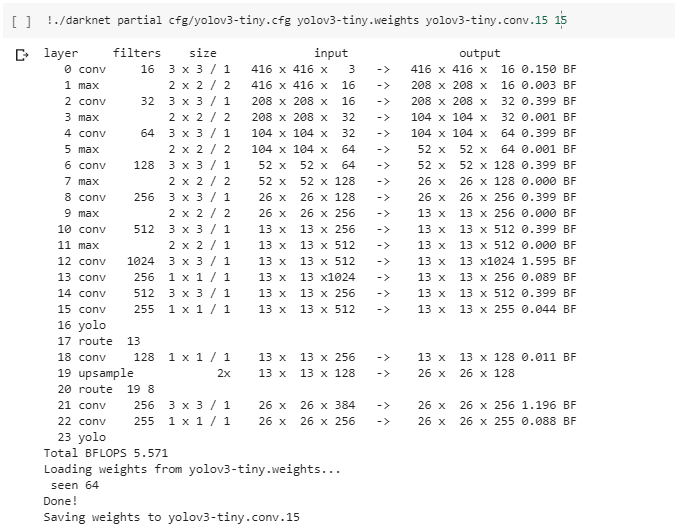
Download default weights file for yolov3-tiny: <https://pjreddie.com/media/files/yolov3-tiny.weights>

Copy **yolov3-tiny.weights** vào thư mục **darknet/**

**yolov3-tiny.conv.15**

Tạo file **yolov3-tiny.conv.15** sử dụng lệnh trên colab

!./darknet partial cfg/yolov3-tiny.cfg yolov3-tiny.weights yolov3-tiny.conv.15 15



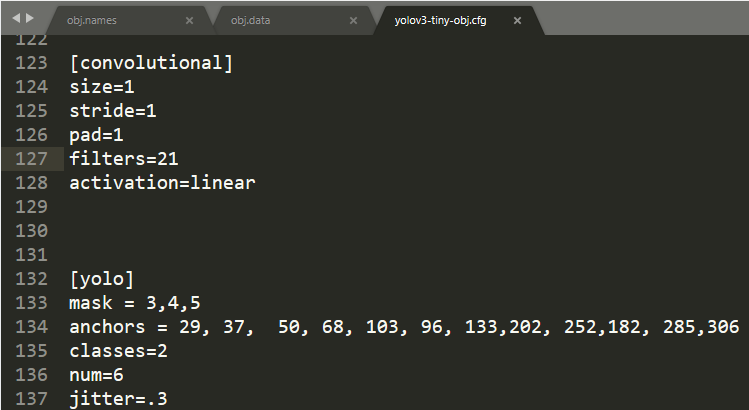
Và file **yolov3-tiny.conv.15** cũng nằm trong thư mục **darknet/**

**yolov3-tiny-obj.cfg**

Tạo file: yolov3-tiny-obj.cfg nằm trong thư mục **darknet/**

từ file (copy): **darknet/cfg/yolov3-tiny\_obj.cfg** instead of **yolov3.cfg**

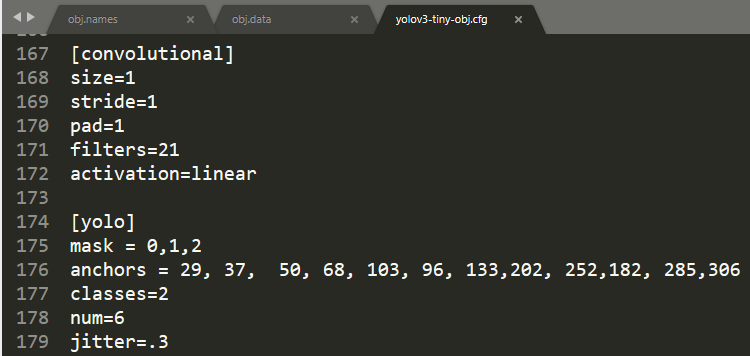
chỉnh sửa nội dung file **yolov3-tiny-obj.cfg**



So if classes=1 then should be filters=18. If classes=2 then write filters=21.

filters=(classes + coords + 1)\*<number of mask>

filters = (2 + 4 + 1)\*3 = 21



Thay đổi anchors:

Sử dụng lệnh:

!./darknet detector calc\_anchors data/obj.data -num\_of\_clusters 6 -width 416 -height 416

Kết quả: 29, 37, 50, 68, 103, 96, 133,202, 252,182, 285,306

Copy bỏ vào mục anchors trong file trên

**Bước 6**: Tiến hành train model

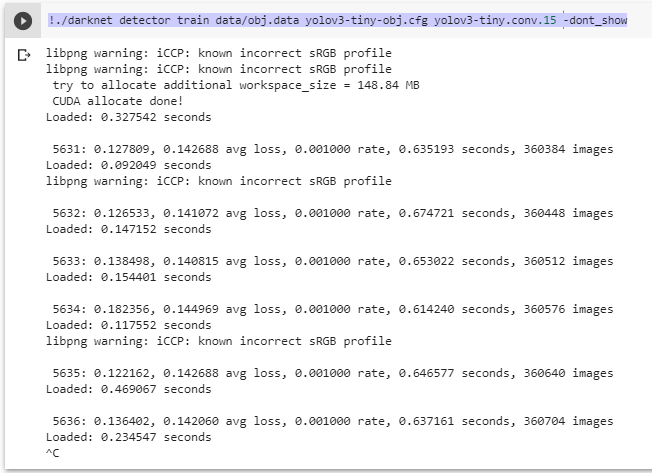
chạy lệnh sau để biến darknet thành file executable nhé:

chmod +x darknet

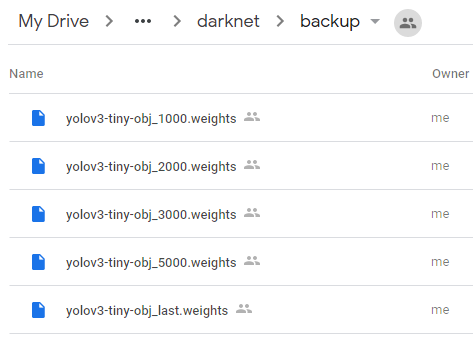
Rồi, cuối cùng chạy lệnh để train nào:

!./darknet detector train data/obj.data yolov3-tiny-obj.cfg yolov3-tiny.conv.15 -dont\_show

Trong quá trình train, bạn để ý 2 tham số loss và avg loss, nếu thấy nó bão hòa và không thay đổi nhiều nữa thì có thể stop lại quá trình train.

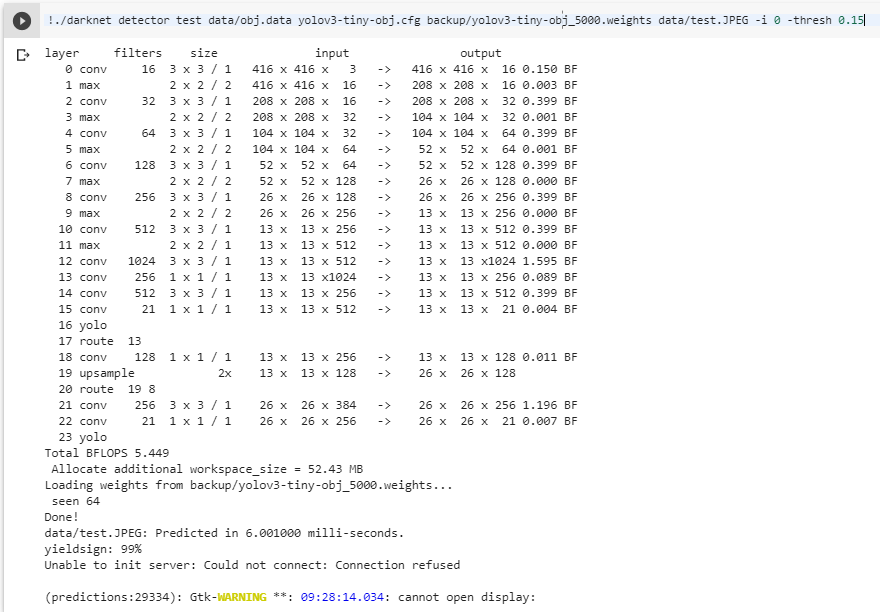


Kết quả các file weights được lưu trong quá trình train

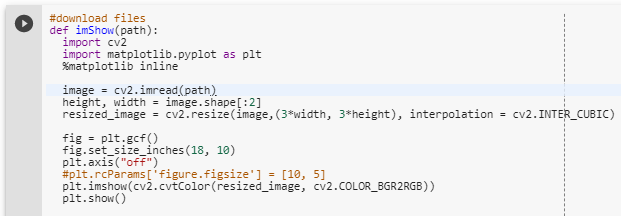


Bước 7. Kiểm thử quá trình train bằng cách detect thử 1 ảnh.

!./darknet detector test data/obj.data yolov3-tiny-obj.cfg backup/yolov3-tiny-obj\_5000.weights data/test.JPEG -i 0 -thresh 0.15

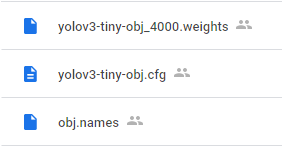


Định nghĩa hàm imShow() để hiển thị kết quả:

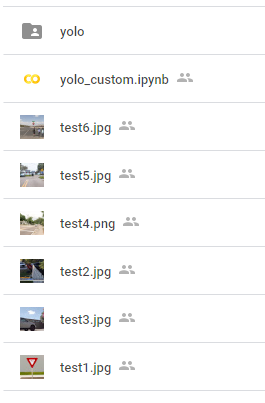


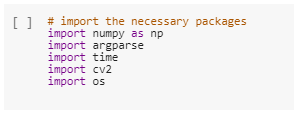


YOLO object detection with OpenCV

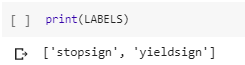
Cấu trúc thư mục:

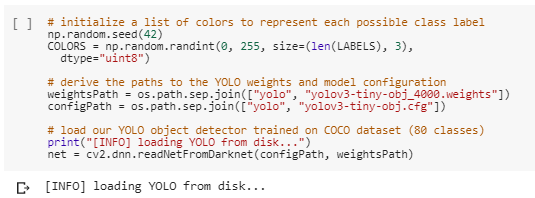
In yolo folder

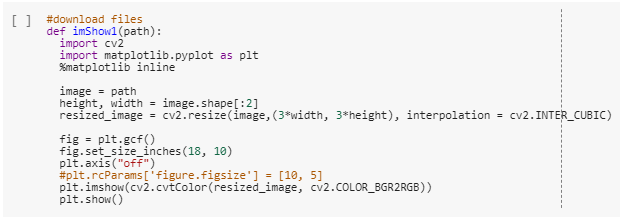


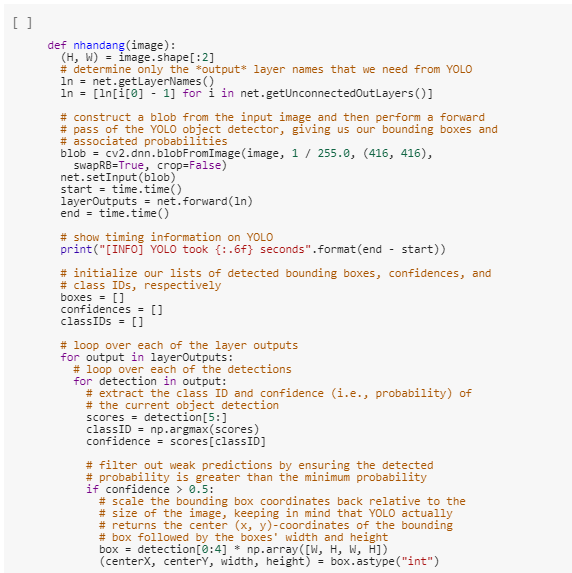
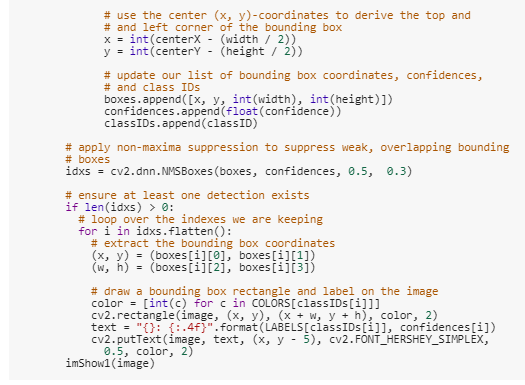




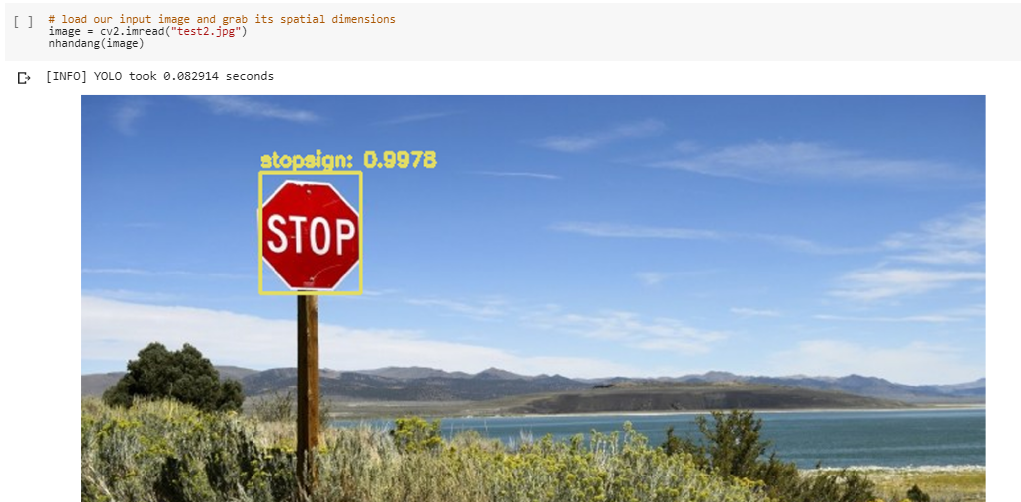


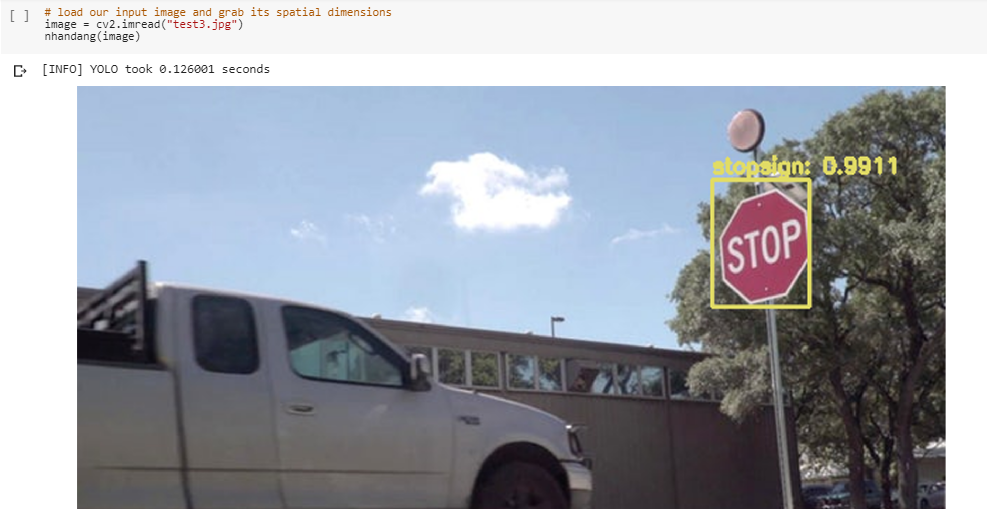


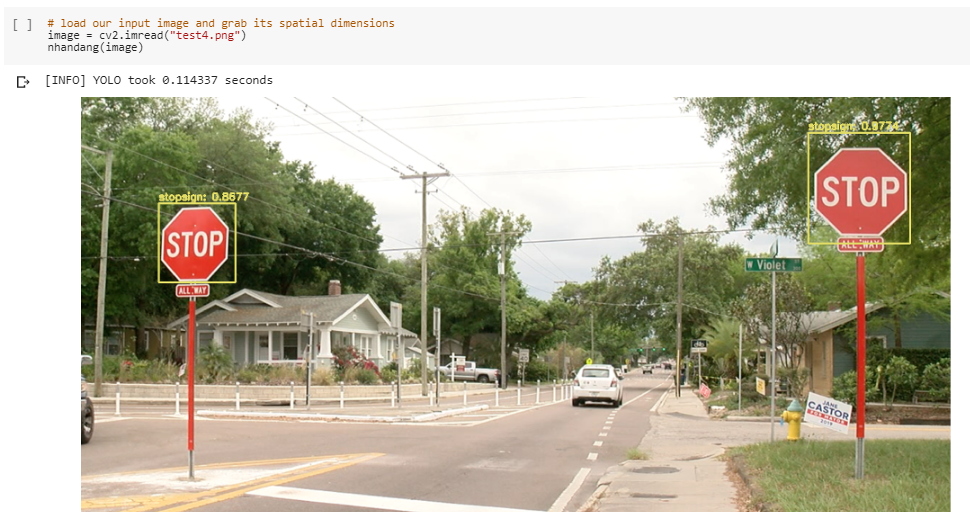












<https://medium.com/@manivannan_data/how-to-train-multiple-objects-in-yolov2-using-your-own-dataset-2b4fee898f17>