For this operation we will need an external software: ****LabelImg.****  
You can download it here for Windows and Linux: <https://tzutalin.github.io/labelImg/>  
If you have the MAC, you can install it by following the instructions here: [https://github.com/tzutalin/labelImg#macOS](https://github.com/tzutalin/labelImg" \l "macOS)

## **2. Train the Image dataset online**

To train the image dataset we’re going to use the free server offered by [google colab](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwjVtorb1MfoAhXFrHEKHXb_BFwQFjAAegQIBxAC&url=https://colab.research.google.com/&usg=AOvVaw3A5aPK2kLFzKOzb6sOckVw).

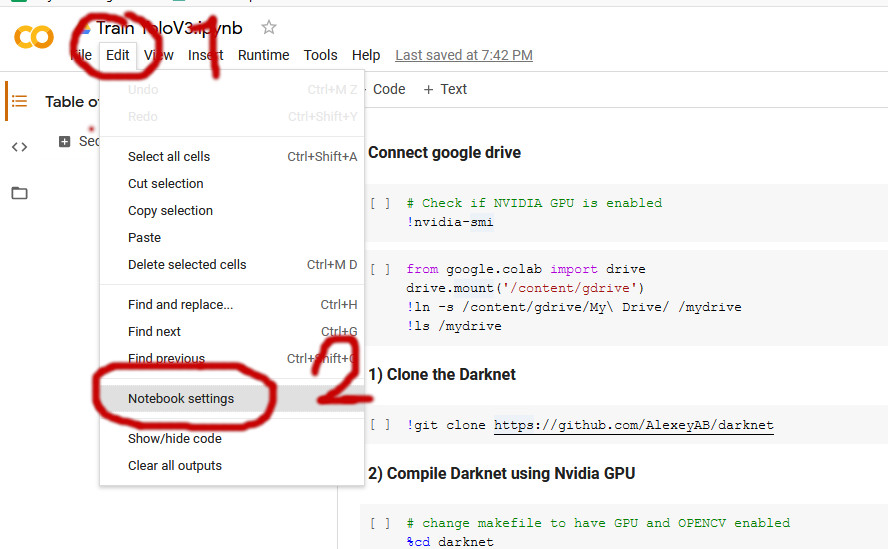
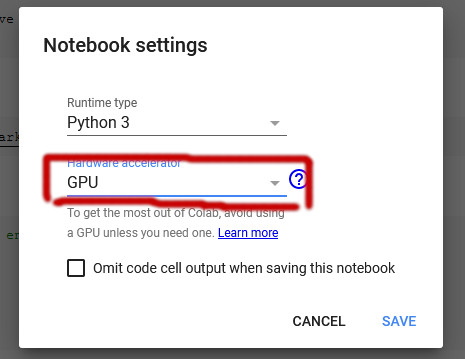
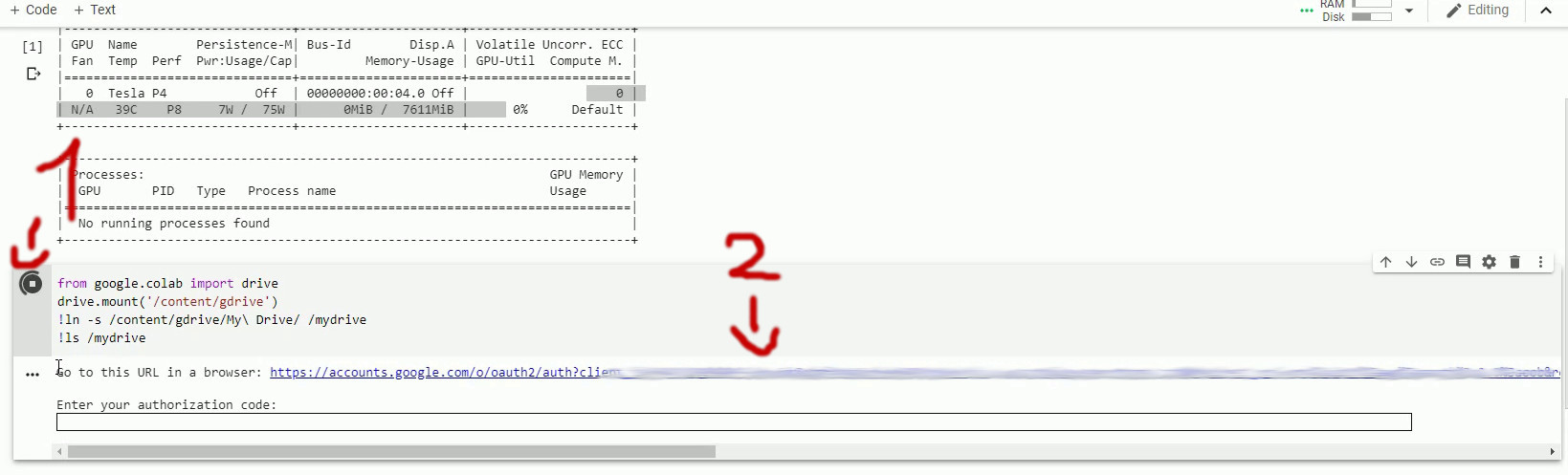
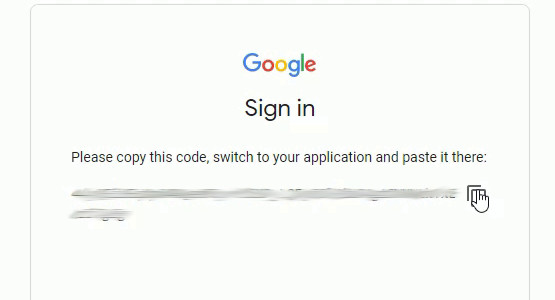
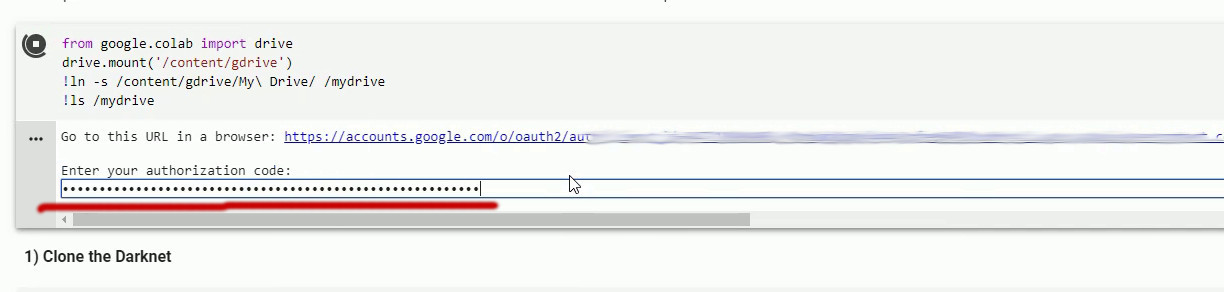
Google colab is a free service offered by google where you can run python scripts and use machine learning libraries taking advantage of their powerful hardware.  
It’s for free with the only disadvantage the you can use it for 12 hours in a row, after that you’ll be disconnected and your files will be deleted.  
you can restart it again but doing everything from scratch.

We can solve this problem by connecting google colab with google drive, so we won’t lose the files in case of disconnection.

****Set up google drive:****

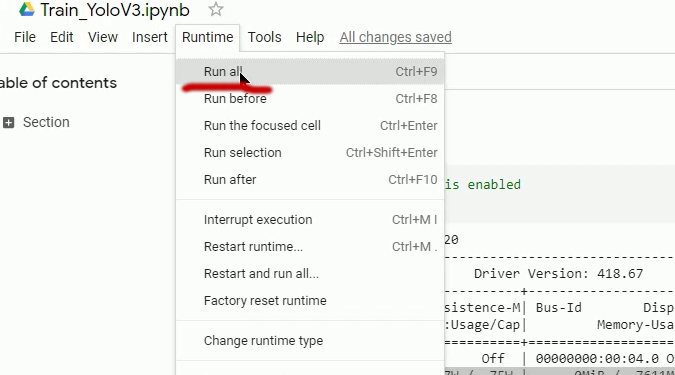
1. Go on [google drive](https://www.google.com/drive/) and log in. If you’ don’t have an account, create one and log in.
2. Create a new folder called “yolov3”.
3. Then upload the file “images.zip” you created before inside the yolov3 folder.

****Set up google colab:****

1. Go on [google colab](https://colab.research.google.com/) and log in with the same account you used to log in on google drive.
2. Upload this file “Train\_YoloV3.ipynb”  
   n.b. You can get this file by clicking on “Click here to download the Source code” at the beginning of the post.
3. Then we need to enable the GPU. So click on “Edit”.  
   
4. Then we select “GPU” and click save.  
   
5. Now we’re ready to connect Colab with our drive.  
   Run the cell where it’s written “from google.colam import drive” and then clicl on the link that appears.  
   
6. This links is to grant access to your google drive. Once you enter it asks you to allow the Google drive file Stream. Click on “Allow”. And then copy the code that appears.  
   
7. Paste the code on the notebook and press “Enter”.  
   

Your colal is ready for the training.

Now you can****start the training**** by simply clicking on “Runtime”, then “Run all”.



The command above is going to run all the cells.  
The code will automatically install the darknet (framework used to run and train YOLO), it will make the configuration and it will run the training.

If you see an output similar to the one below, then well done, your model is training.



Leav it as long as it goes. Usually it takes a minimum of 3 hours to the entire 12 hours you have the access to the gpu.

We will see now how to use the model you’ve created.

## **3. Test the model we created**

Each 100 iterations, our custom object detector is going to be updated and saved on our Google drive, inside the folder “yolov3”.

The ****file that we need is “yolov3\_training\_last.weights”.****  
You might find that other files are also saved on your drive, “yolov3\_training\_\_1000.weights”, “yolov3\_training\_2000.weights” and so on because the darknet makes a backup of the model each 1000 iterations.

I created a python project to test your model with Opencv.  
The project is on the folder yolo\_custom\_detection, which contains 2 files (yolo\_object\_detection.py and yolov3\_testing.cfg).  
n.b. You can get this file by clicking on “Click here to download the Source code” at the beginning of the post.

You need to download the file ****yolov3\_training\_last.weights**** from Google Drive and place in on the same folder with yolo\_object\_detection.py and yolov3\_testing.cfg.  
  
On the file “yolo\_object\_detection.py” on line 11 change “koala” with the name of your object.  
And on line 14 you have to put the path of a folder containing the images where there is your object, so you can test it.

# Load Yolo

net = cv2.dnn.readNet("yolov3\_training\_last.weights", "yolov3\_testing.cfg")

# Name custom object

classes = ["Koala"]

# Images path

images\_path = glob.glob(r"D:\Pysource\Youtube205) Train Yolo google cloud\dataset\\*.jpg")

And finally, here is the result.  
It works!

