**Setting up darknet on Linux**

**CuDNN**

Get cudnn:

<https://docs.nvidia.com/deeplearning/cudnn/latest/installation/linux.html>

| sudo apt-get -y install cudnn9-cuda-12 |
| --- |

**Check if the installation is correct:**

<https://docs.nvidia.com/deeplearning/cudnn/latest/installation/linux.html#verifying-the-install-on-linux>

In case you can’t find cudnn\_samples use **sudo find**:

e.g.: sudo find / -name Taxes-2013.pdf

If there is any error regarding freeimage during make install it with the following command:

| sudo apt-get install libfreeimage3 libfreeimage-dev |
| --- |

**OpenCV**

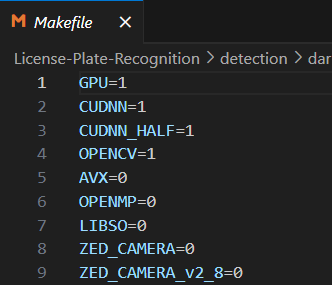
| sudo apt update sudo apt install libopencv-dev |
| --- |

**Darknet**

Get darknet repo:

| git clone https://github.com/AlexeyAB/darknet |
| --- |

In the makefile set GPU, CUDNN and OPENCV to 1:



Go to darknet and build:

| cd darknet make |
| --- |

If darknet is already built but for some reason you want to rebuild it, use the flag -B with the make command.

Get yolov4 weights:

| wget https://github.com/AlexeyAB/darknet/releases/download/darknet\_yolo\_v3\_optimal/yolov4.weights |
| --- |

Test the model on the sample image:

| ./darknet detect cfg/yolov4.cfg yolov4.weights data/dog.jpg |
| --- |

You should see the results in predictions.jpg

**Darknet data organization**

Dataset must be organized in the following way: for each image there must exist a text file, named the same as the image, with the annotations. In the annotations file, each object is defined on a new line mentioning the following:

**classid, x, y, w, h**

where (x, y) is the center of the founding box (normalized to the width and height of the image) and (w, h) are dimensions of the bounding box, normalized as well. Example:

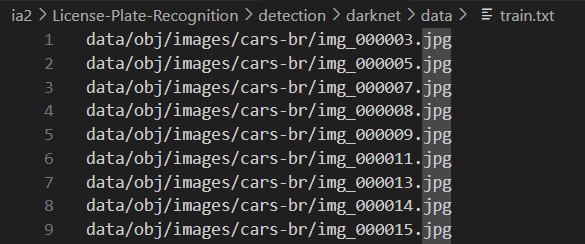


For training with darknet, the following files must be created:

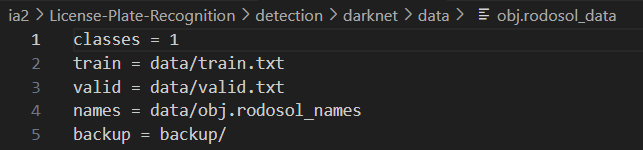
* **obj.names** - text file containing the names of the classes in the dataset. The line number of each class represents its class id. Example from rodosol dataset (has only one class: plate):



* **train.txt, test.txt/valid.txt** - txt files containing paths to the images relative to the darknet directory.



* **yolo.cfg**, configuration file of the model - darknet repository has a bunch of examples in the cfg directory.
* **obj.data**, contains the number of classes, paths to train and test txt files, class names and the output directory where models will be saved (backup)



./darknet detector train data/obj.rodosol\_data cfg/yolov4-custom-rodosol.cfg yolov4.conv.137 -dont\_show -mjpeg\_port 8090 -map

darknet detector test /home/ia2/License-Plate-Recognition/detection/darknet/data/obj.rodosol\_data /home/ia2/License-Plate-Recognition/detection/darknet/cfg/yolov4-custom-rodosol.cfg /home/ia2/License-Plate-Recognition/detection/darknet/backup\_combined\_pretrained/yolov4-custom-rodosol\_best.weights -ext\_output -dont\_show -out result\_combined\_rodosol.json < /home/ia2/License-Plate-Recognition/detection/darknet/data/test.txt