**This document is included as an extension of the explanatory text of the project: https://github.com/ablanco1950/DetectTrafficSign**

**Creation of a model based on yolov8 with custom dataset https://www.kaggle.com/datasets/valentynsichkar/traffic-signs-dataset-in-yolo-format/data**

Download from <https://www.kaggle.com/datasets/valentynsichkar/traffic-signs-dataset-in-yolo-format/data> the Kaggle files that will be used for training by yolov8:

You will get an archive(12).zip file that, decompressed, becomes an archive(12) with a ts directory followed by another ts in which .jpg files are found along with others with the same name and .txt extension,

Yolov8 requires that in the same directory of the program that executes the train contains the train folders, with the subfolders images and labels and valid with the subfolders images and labels.

Images can be moved from the download ts directory to the train images subfolder by selecting \*.jpg files from ts and copying and pasting over train images.

In the same way, the \*.txt files from ts are selected and copied and pasted onto the train labels.

Next, 30% of the train images are selected, selecting the last 245 images of train images, from 00560.jpg at the end, they are cut and pasted over the directory images of the valid directory

In the same way, 30% of the train labels are selected, selecting the last 245 train labels, from 00560.txt to the end, cut and pasted on the directory labels of the valid directory

Once the directory structure required by yolov8 has been built, the training process begins by executing:

DetectTrafficSignYolov8Train.py

Running from a simple laptop, the 100 epochs of the program will take a long time, but you can always lower the cover of the laptop and continue the next day.

The log of the process can be consulted in the file LOG\_TRAIN\_DetectTrafficSign.txt

As a result, inside the project folder, the directory runs\detect\trainN\weights( where in trainN, N indicates the last train directory created, in which the best.pt file is located), best.pt is the base of the model .

References:

The steps followed are analogous to those used to detect car license plates in the project:

https://github.com/ablanco1950/LicensePlate\_Yolov8\_Filters\_PaddleOCR