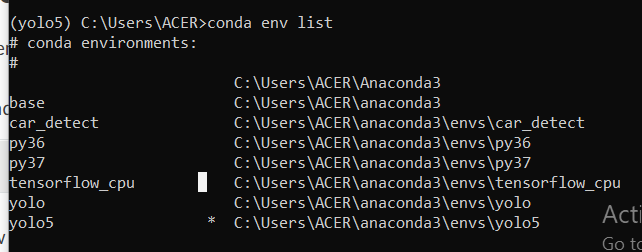
**Step 1** ( very important ): Install required package and module

This project run the virtual conda environment named yolo5



Every requirement for version of libraries and packages and modules are specified in the file requirements.txt:

# pip install -U -r requirements.txt  
Cython  
numpy==1.17  
opencv-python  
torch>=1.4  
matplotlib  
pillow  
tensorboard  
PyYAML>=5.3  
torchvision  
scipy  
tqdm  
git+https://github.com/cocodataset/cocoapi.git#subdirectory=PythonAPI

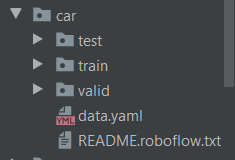
To install all of these, just run “pip install -U -r requirements.txt”

Step 2: Prepare data

Look at the folder BCCD, we need to prepare data for folders train, valid because we use them for training, and folder test to contain data for testing. One tip is that valid data would have our model to increase the accuracy after training.

Step 3: Adjust the config file

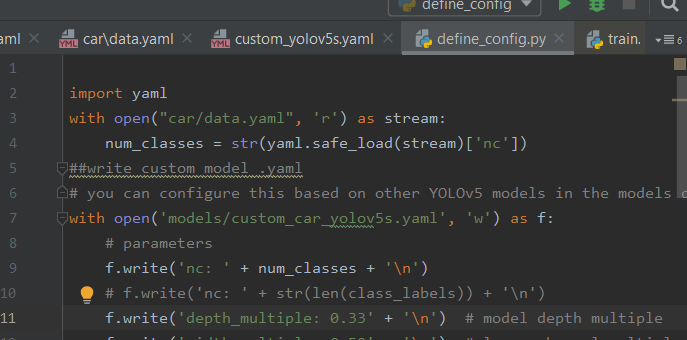
With each dataset, it would have a file data.yaml



We have to create another file yaml basing on this data.yaml file

To do that, we need to parse the path of our data.yaml file to file define\_config.py and define the name of output yaml file in define\_config.py to run it.

For example, with our example:



In the file <name of folder containing data>/data.yaml ( for example , BCCD/data.yaml), we need to provide corresponding data of our dataset.

Step 4: Training

In this project, we can run this line in the terminal of pycharm to train:

python train.py --img 416 --batch 16 --epochs 100 --data BCCD/data.yaml --cfg models/custom\_yolov5s.yaml --name yolov5s\_results --nosave --cache

But, with new data, we have to adjust some numbers there.

Step 5: Testing

After training, the weight would be saved in directory weights

For example, in our project, it is weights/last\_yolov5s\_results.pt

Then, for this project, if we want to run, we can run this on the terminal:

python detect.py --weights weights/last\_yolov5s\_results.pt --img 416 --conf 0.4 --source inference/images

here, the folder inference/images would contain images for testing.

And running this would output images with predicted boxes in the folder inference/output