

```
!git clone https://github.com/ultralytics/yolov5 # clone
%cd yolov5
!pip install -qr requirements.txt # install

import torch
import utils
display = utils.notebook_init() # checks
```

YOLOv5 v6.2-205-geef9057 Python-3.7.15 torch-1.12.1+cu113 CUDA:0 (Tesla T4, 15110MiB)
Setup complete (2 CPUs, 12.7 GB RAM, 38.8/78.2 GB disk)

- 1.) -ทำการ git clone yolov5
-import torch และ library ที่ใช้(utils)

```
[ ] !python detect.py --weights yolov5s.pt --conf 0.25 --source data/images
#display.Image(filename='runs/detect/exp/zidane.jpg', width=600)
```

detect: weights=['yolov5s.pt'], source=data/images, data=data/coco128.yaml, imgsz=[640, 640], conf_thres=0.25, iou_thres=0.45
YOLOv5 v6.2-205-geef9057 Python-3.7.15 torch-1.12.1+cu113 CUDA:0 (Tesla T4, 15110MiB)

Downloading <https://github.com/ultralytics/yolov5/releases/download/v6.2/yolov5s.pt> to yolov5s.pt...
100% 14.1M/14.1M [00:00<00:00, 297MB/s]

Fusing layers...
YOLOv5s summary: 213 layers, 7225885 parameters, 0 gradients
image 1/2 /content/yolov5/data/images/bus.jpg: 640x480 4 persons, 1 bus, 17.7ms
image 2/2 /content/yolov5/data/images/zidane.jpg: 384x640 2 persons, 2 ties, 12.6ms
Speed: 0.5ms pre-process, 15.2ms inference, 26.9ms NMS per image at shape (1, 3, 640, 640)
Results saved to **runs/detect/exp**

- (optional) test yolov5 ด้วยการ run detect รูป จากไฟล์ที่มีใน yolov5 ที่โคลนมา
(if) ไม่สำเร็จให้ทำการโคลนใหม่ กลับไปตอน 1
ผลลัพธ์อยู่ใน **yolov5/runs/detect/exp**

Train

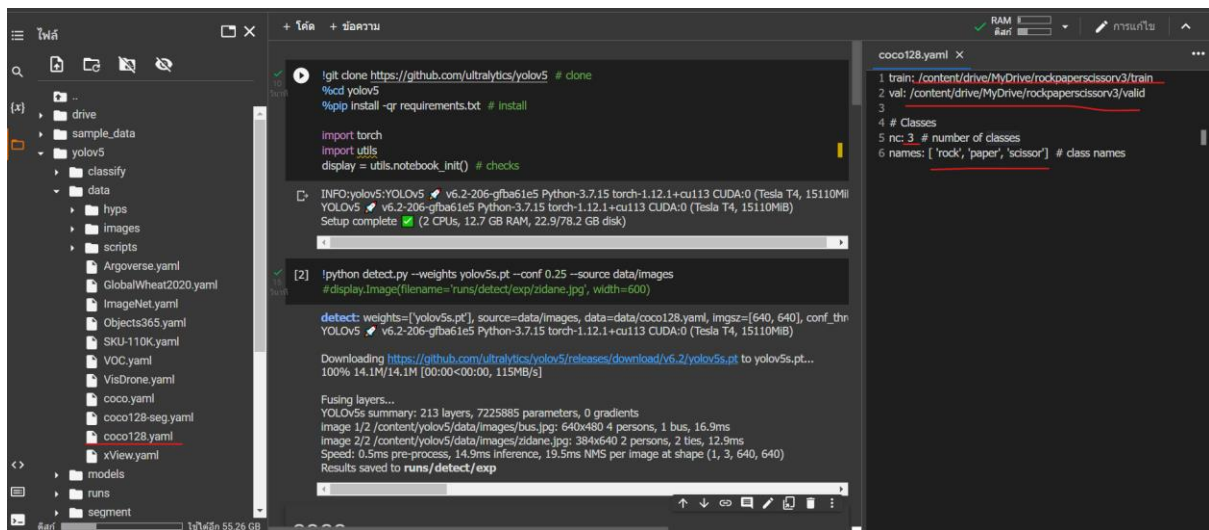
```
%pip install -q wandb
import wandb
wandb.login()
```

1.9 MB	14.0 MB/s
166 kB	63.1 MB/s
182 kB	65.5 MB/s
63 kB	1.7 MB/s
166 kB	66.9 MB/s
162 kB	66.9 MB/s
162 kB	70.1 MB/s
158 kB	69.9 MB/s
157 kB	67.6 MB/s
157 kB	70.9 MB/s
157 kB	65.9 MB/s
157 kB	73.7 MB/s
157 kB	74.5 MB/s
157 kB	76.0 MB/s
157 kB	70.4 MB/s
156 kB	69.6 MB/s

Building wheel for pathtools (setup.py) ... done
Failed to detect the name of this notebook, you can set it manually with the WANDB_NOTEBOOK_NAME environment variable to enable code saving.
wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc
True

- (optional) login เข้า wandb เพื่อ visualize การเทรนหลังจากเทรนเสร็จ
ครั้งแรกหน้านี้จะถามหาโค้ดให้ทำการสมัครแล้วนำโค้ดมารอกที่ช่องว่างในหน้านี้

Setup



2.) -หาไฟล์ coco128.yaml ลบข้อมูลข้างในทั้งหมดแล้วแก้ไขไฟล์เป็นตามนี้(คลิกสองครั้งสามารถเข้าไปแก้ไขได้)

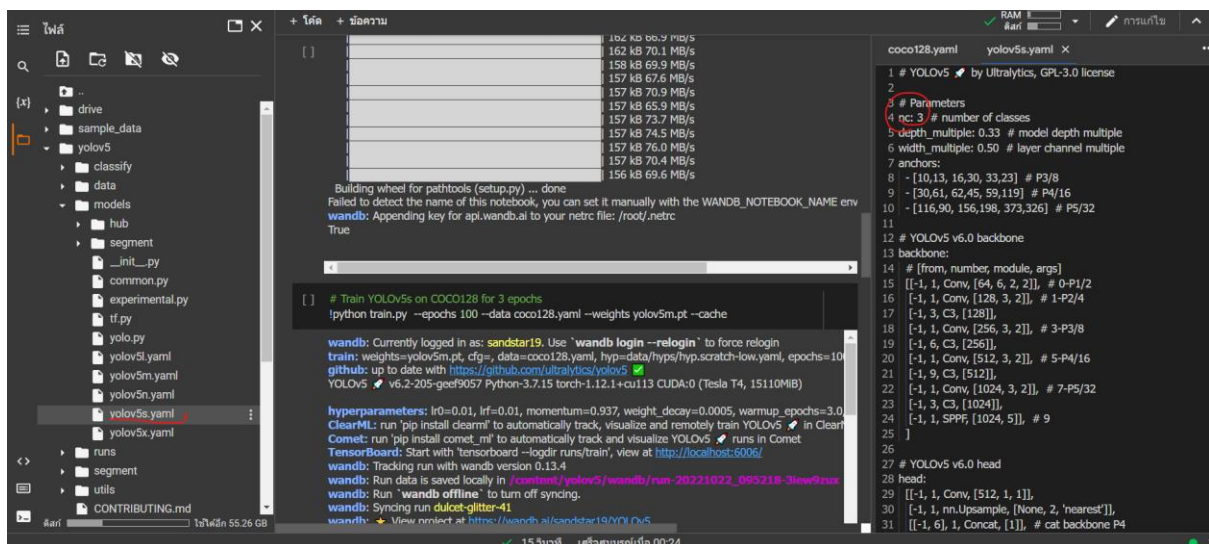
train: /content/drive/MyDrive/rockpaperscissorv3/train #path train ของไฟล์ที่เก็บรูปไว้ในgoogle drive

val: /content/drive/MyDrive/rockpaperscissorv3/valid #path valid ของไฟล์ที่เก็บรูปไว้ในgoogle drive

Classes

nc: 3 # number of classes จำนวนของคลาสที่จะจำแนก

names: ['rock', 'paper', 'scissor'] # class names ชื่อของคลาสที่จะแยกหรือตรวจจับเขียนในรูปarray



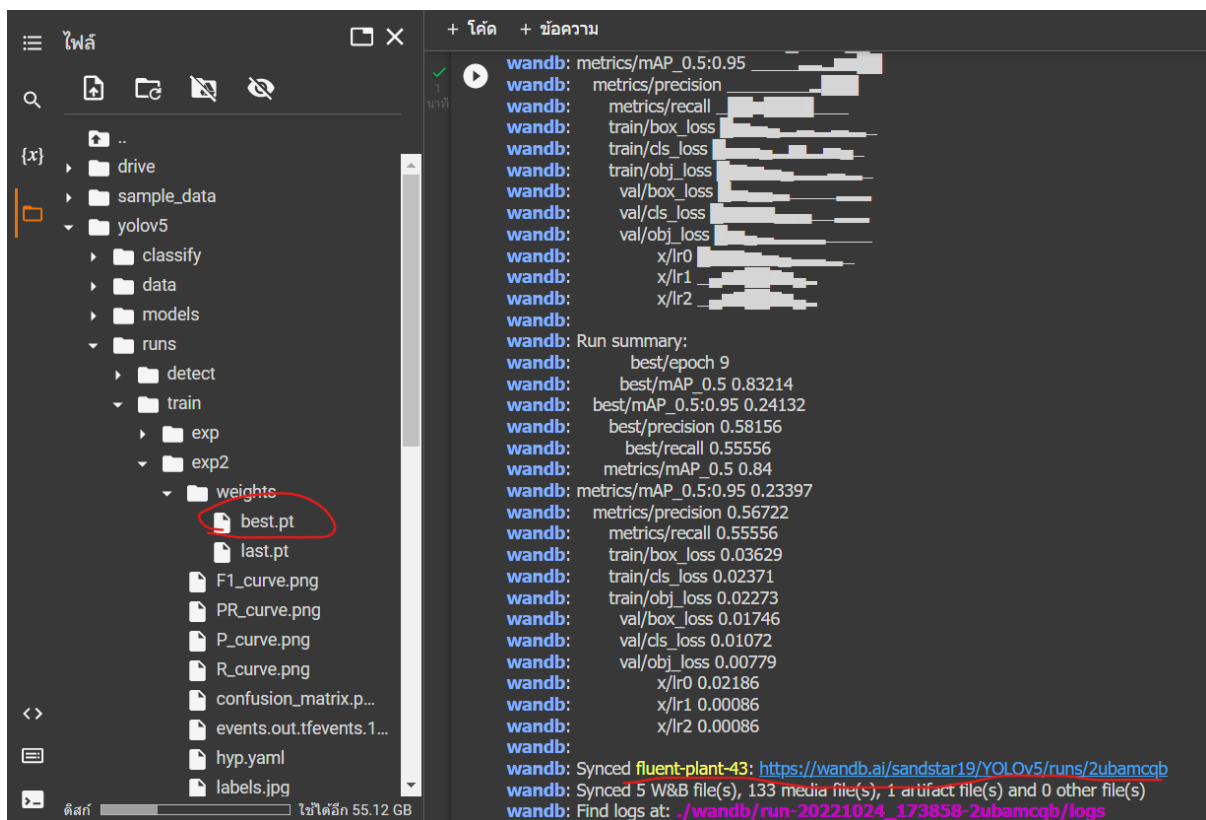
3.) -หาไฟล์ yolov5s.yaml แก้ nc: ให้เป็นจำนวนคลาสที่เราจะจำแนก

```
# Train YOLOv5s on COCO128 for 3 epochs
!python train.py --epochs 10 --data coco128.yaml --weights yolov5s.pt --cache

wandb: Currently logged in as: sandstar19. Use `wandb login --relogin` to force relogin
train: weights=yolov5s.pt, cfg=, data=coco128.yaml, hyp=data/hyps/hyp.scratch-low.yaml, epochs=10, batch_size=16, imgsz=640
github: up to date with https://github.com/ultralytics/yolov5
YOLOv5 v6.2-206-gfba61e5 Python-3.7.15 torch-1.12.1+cu113 CUDA:0 (Tesla T4, 15110MiB)

hyperparameters: lr0=0.01, lrf=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8,
ClearML: run 'pip install clearml' to automatically track, visualize and remotely train YOLOv5 in ClearML
Comet: run 'pip install comet_ml' to automatically track and visualize YOLOv5 runs in Comet
TensorBoard: Start with 'tensorboard --logdir runs/train', view at http://localhost:6006/
wandb: Tracking run with wandb version 0.13.4
wandb: Run data is saved locally in /content/yolov5/wandb/run-20221024_173858-2ubamcqb
wandb: Run `wandb offline` to turn off syncing.
wandb: Syncing run fluent-plant-43
wandb: View project at https://wandb.ai/sandstar19/YOLOv5
wandb: View run at https://wandb.ai/sandstar19/YOLOv5/runs/2ubamcqb
Overriding model.yaml nc=80 with nc=3
```

- 4.) -ทำการเริ่มเทรน โดยกำหนด epoch/data/weight มาเทรน แนะนำที่ 100-200 epoch (ยิ่งหลาย epoch ยิ่งใช้เวลานาน)



```
wandb: metrics/mAP_0.5:0.95
wandb: metrics/precision
wandb: metrics/recall
wandb: train/box_loss
wandb: train/cls_loss
wandb: train/obj_loss
wandb: val/box_loss
wandb: val/cls_loss
wandb: val/obj_loss
wandb: x/lr0
wandb: x/lr1
wandb: x/lr2
wandb:
wandb: Run summary:
wandb: best/epoch 9
wandb: best/mAP_0.5 0.83214
wandb: best/mAP_0.5:0.95 0.24132
wandb: best/precision 0.58156
wandb: best/recall 0.55556
wandb: metrics/mAP_0.5 0.84
wandb: metrics/mAP_0.5:0.95 0.23397
wandb: metrics/precision 0.56722
wandb: metrics/recall 0.55556
wandb: train/box_loss 0.03629
wandb: train/cls_loss 0.02371
wandb: train/obj_loss 0.02273
wandb: val/box_loss 0.01746
wandb: val/cls_loss 0.01072
wandb: val/obj_loss 0.00779
wandb: x/lr0 0.02186
wandb: x/lr1 0.00086
wandb: x/lr2 0.00086
wandb:
wandb: Synced fluent-plant-43: https://wandb.ai/sandstar19/YOLOv5/runs/2ubamcqb
wandb: Synced 5 W&B file(s), 133 media file(s), 1 artifact file(s) and 0 other file(s)
wandb: Find logs at: ./wandb/run-20221024_173858-2ubamcqb/logs
```

- 5.) -เมื่อเทรนเสร็จให้ไปที่expล่าสุดแล้วโหลด best.pt ไปใช้ได้เลย(เอาไปใส่โฟลเดอร์ yolov5ในคอม)

-Link wandb สามารถกดเข้าไปดูresultหารเทรนได้