YOLOX-S LINUX COMMANDS

Train:

python3 tools/train.py -f exps/example/yolox\_voc/yolox\_voc\_s.py -d 1 -b 1 --fp16 -c /home/imjaekyu/Desktop/YOLOX/yolox\_s.pth

Camera:

python3 tools/demo2.py webcam ~~-~~f exps/example/yolox\_voc/yolox\_voc\_s.py -c best\_ckpt.pth --camid 0 --conf 0.5 --nms 0.1 --tsize 640 --device gpu

Camera with TensorRT:

python3 tools/demo2.py webcam -f exps/example/yolox\_voc/yolox\_voc\_s.py -c model\_trt.pth --camid 0 --conf 0.5 --nms 0.1 --tsize 640 --trt --device gpu

Demo Video: **(Only work on default demo.py file)**

python tools/demo.py video -f exps/example/yolox\_voc/yolox\_voc\_s.py -c epoch\_160\_ckpt.pth --path assets/1.mp4 --conf 0.1 --nms 0.1 --tsize 640 --save\_result --device gpu

Demo Picture:

python3 tools/demo2.py image -f exps/example/yolox\_voc/yolox\_voc\_s.py -c best\_ckpt.pth --path assets/1.jpg --conf 0.5 --nms 0.1 --tsize 640 --save\_result --device gpu

TensorRT Converter:

python3 tools/trt.py -f exps/example/yolox\_voc/yolox\_voc\_s.py -c epoch\_160\_ckpt.pth

**(the result of the conversion is model\_trt.pth)**

**(If you want to see the conversion progress, delete model\_trt.engine and model\_trt.pth in yolox\_voc\_s folder and run the linux command in the terminal)**

**camid = Camera ID**

**fp16 = Floating Point 16**

**save\_result = Save the result of a resulted image or video**

**device = GPU / CPU**

**trt = TensorRT**

**conf = Confidence Level**

**nms = Non-Max Suppression**

**d = Device (For multiple GPU, then d will be more than 1)**

**b = Batch Size (Recommended below 32, If training dataset is low, we should lower the batch size to 4 or 8)**