

YOLOv4 on Windows Subsystem for Linux

Ubuntu 18.04 on WSL

Linux kernel version: 5.10.60.1-microsoft-standard-WSL2

GPU Driver version: 510.06

CUDA version: 11.0

CPU: i7-9750H

GPU: RTX 2070 8G GDDR6 Max-Q

RAM: 32G DDR4 2666

DISK: RAID 0

Install darknet from AlexeyAB's github

Github: <https://github.com/AlexeyAB/darknet>

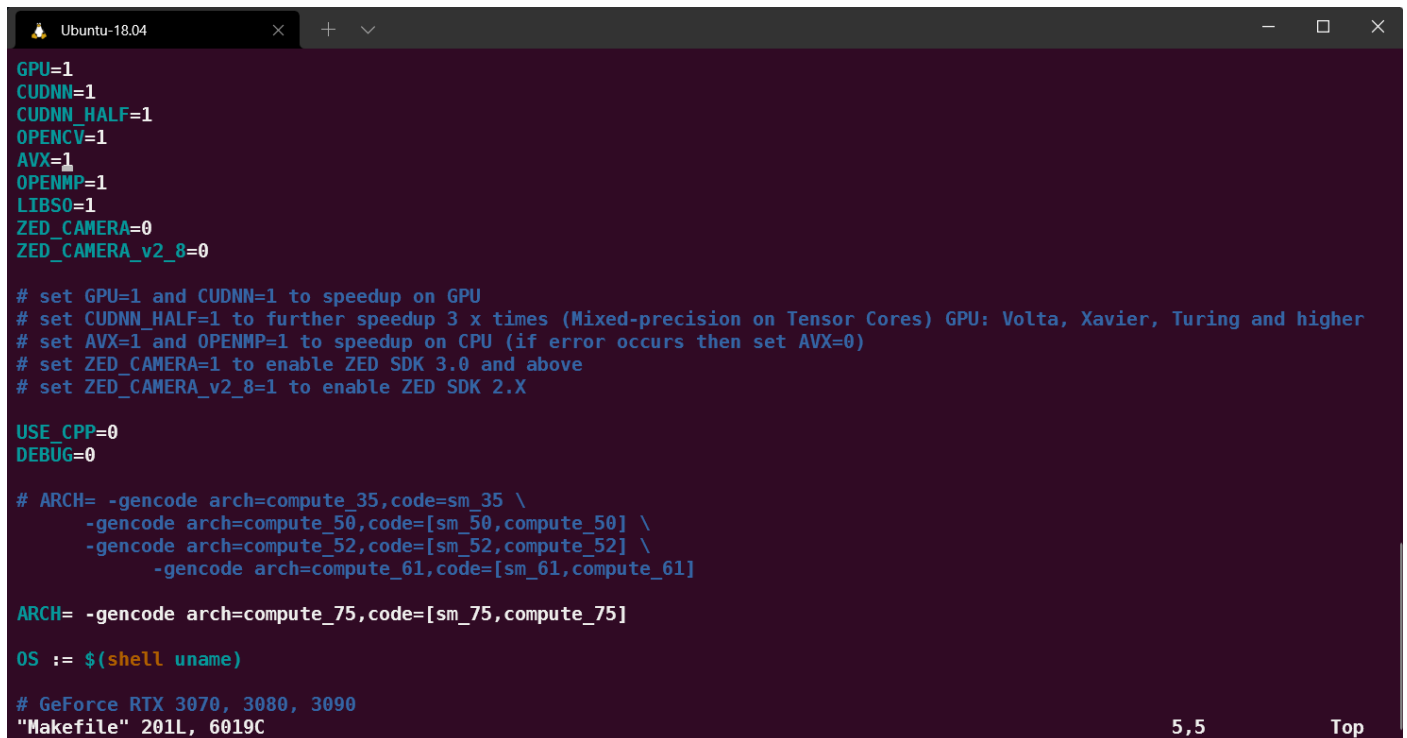
Reference: <https://www.rs-online.com/designspark/nvidia-jetson-nanotensor-rtyolov4-cn>

Clone from Github

```
$ git clone https://github.com/AlexeyAB/darknet.git
```

```
$ cd darknet
```

Modify Makefile



```
GPU=1
CUDNN=1
CUDNN_HALF=1
OPENCV=1
AVX=1
OPENMP=1
LIBS0=1
ZED_CAMERA=0
ZED_CAMERA_v2_8=0

# set GPU=1 and CUDNN=1 to speedup on GPU
# set CUDNN_HALF=1 to further speedup 3 x times (Mixed-precision on Tensor Cores) GPU: Volta, Xavier, Turing and higher
# set AVX=1 and OPENMP=1 to speedup on CPU (if error occurs then set AVX=0)
# set ZED_CAMERA=1 to enable ZED SDK 3.0 and above
# set ZED_CAMERA_v2_8=1 to enable ZED SDK 2.X

USE_CPP=0
DEBUG=0

# ARCH= -gencode arch=compute_35,code=sm_35 \
#         -gencode arch=compute_50,code=[sm_50,compute_50] \
#         -gencode arch=compute_52,code=[sm_52,compute_52] \
#         -gencode arch=compute_61,code=[sm_61,compute_61]

ARCH= -gencode arch=compute_75,code=[sm_75,compute_75]

OS := $(shell uname)

# GeForce RTX 3070, 3080, 3090
"Makefile" 201L, 6019C
```

CUDNN_HALF: Enable support Tensor Cores (RTX GPU)

OPENCV: Allow to detect on video files and video streams. (Support OpenCV 4.X/3.X/2.4.X)

```
// Install OpenCV
$ sudo apt update
$ sudo apt install libopencv-dev python3-opencv
// This will install older version of OpenCV from Ubuntu repository
```

```
erebus@9S7-16Q411-1023 ~
$ pkg-config --modversion opencv
3.2.0
```

```
// Check OpenCV version
```

AVX: Enable AVX support (Improve data augmentation speed)

```
// Check CPU flag for AVX support
```

```
Ubuntu-18.04  x  Ubuntu-18.04  x  +  v
erebus@9S7-16Q411-1023 ~
$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                12
On-line CPU(s) list:   0-11
Thread(s) per core:    2
Core(s) per socket:    6
Socket(s):             1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 158
Model name:            Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz
Stepping:              10
CPU MHz:               2591.991
BogoMIPS:              5183.98
Virtualization:        VT-x
Hypervisor vendor:     Microsoft
Virtualization type:   full
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              12288K
Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 s
s ht syscall nx pdpe1gb rdtscp lm constant_tsc rep_good nopl xtopology cpuid pni pclmulqdq vmx ssse3 fma cx16 pcid sse4_
1 sse4_2 movbe popcnt aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch invpcid_single pti ssbd ibrs ibpb s
tibp tpr_shadow vnmi ept vpid ept_ad fsgsbase bml avx2 smep bmi2 erms invpcid rdseed adx smap clflushopt xsaveopt xsave
c xgetbv1 xsaves flush_l1d arch_capabilities
```

OPENMP: Enable OpenMP support

```
$ make
```

```
// Troubleshooting
```

When making the darknet, an error `‘/usr/bin/ld: cannot find -lcuda’` popped out

Fix by link `/usr/local/cuda/lib64/stubs/libcuda.so` to `/usr/local/cuda/lib64`

Check compile result

```
erebus@9S7-16Q411-1023 ~/darknet
$ ./darknet detector --help
CUDA-version: 11000 (11060), cuDNN: 8.0.4, CUDNN_HALF=1, GPU count: 1
CUDNN_HALF=1
OpenCV version: 3.2.0
usage: ./darknet detector [train/test/valid/demo/map] [data] [cfg] [weights (optional)]
```

Download YOLOv4 weights

https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4.weights

Download YOLOv4-tiny weights

https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v4_pre/yolov4-tiny.weights

Test on Image

```
$ ./darknet detector test ./cfg/coco.data ./cfg/yolov4.cfg ./yolov4.weights
```



Run on Video (limitation)

```
$ ./darknet detector demo cfg/coco.data cfg/yolov4.cfg yolov4.weights sample.mp4 -out_filename sample_.mp4
```