10.YOLOv4-tiny environment building and camera real-time detection

Due to the relatively small memory of the 2GB version, running yolo4 will be stuck. The difference between yolov4 and yolov4-tiny is: yolov4-tiny is a compressed version of yolov4, mainly running the small computing power cpu core version, and using the tiny version frame on Jetson NANO. The rate will be more than ten times higher than that of yolov4.

In generally, it is recommended to use yolov4-tiny on Jetson NANO 2G.

1. Install CUDA，OpenCV，cuDNN

Please refer to [1.Preparation tutorial] for details.

2. Download

```
git clone https://github.com/pjreddie/darknet.git
```

3. Configuration

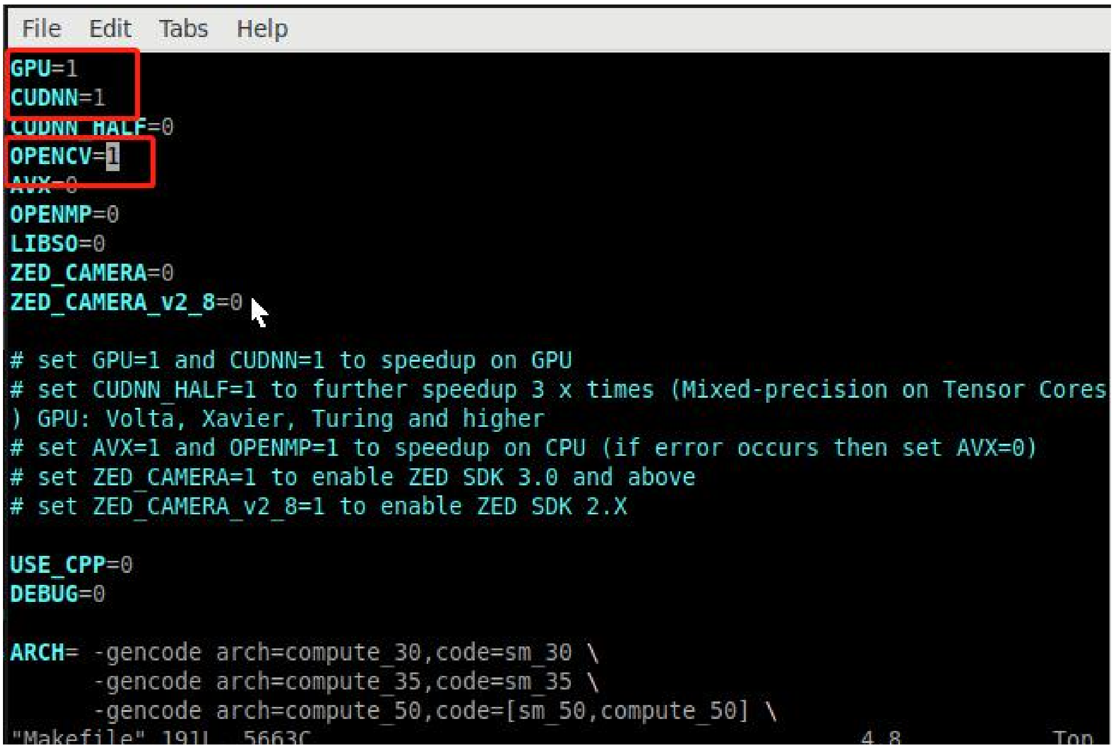
```
cd darknet
sudo vim Makefile      #Modify Makefile
```

4. Modify the first three lines of the Makefile

GPU=1

CUDNN=1

OPENCV=1

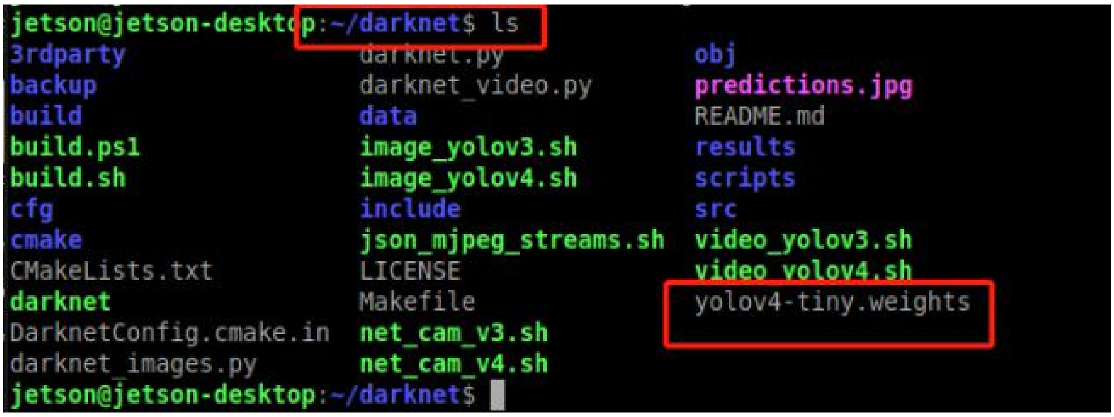


```
GPU=1
CUDNN=1
CUDNN_HALF=0
OPENCV=1
AVX=0
OPENMP=0
LIBSO=0
ZED_CAMERA=0
ZED_CAMERA_v2_8=0

# 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_30,code=sm_30 \
      -gencode arch=compute_35,code=sm_35 \
      -gencode arch=compute_50,code=[sm_50,compute_50] \
"Makefile" 191L, 5663C                                        4,8           Top
```

5. Compile

```
make    -j4
```

6. Copy the weight file yolov4-tiny.weights to the darknet directory.



```
jetson@jetson-desktop:~/darknet$ ls
3rdparty                  darknet.py              obj
backup                    darknet_video.py        predictions.jpg
build                     data                    README.md
build.ps1                 image_yolov3.sh         results
build.sh                  image_yolov4.sh         scripts
cfg                       include                 src
cmake                     json_mjpeg_streams.sh   video_yolov3.sh
CMakeLists.txt            LICENSE                 video_yolov4.sh
darknet                   Makefile                yolov4-tiny.weights
DarknetConfig.cmake.in    net_cam_v3.sh
darknet_images.py         net_cam_v4.sh
jetson@jetson-desktop:~/darknet$
```

7. Test

Yolov4-tiny picture detection

./darknet detect cfg/yolov4-tiny.cfg yolov4-tiny.weights data/dog.jpg   # Shorthand version

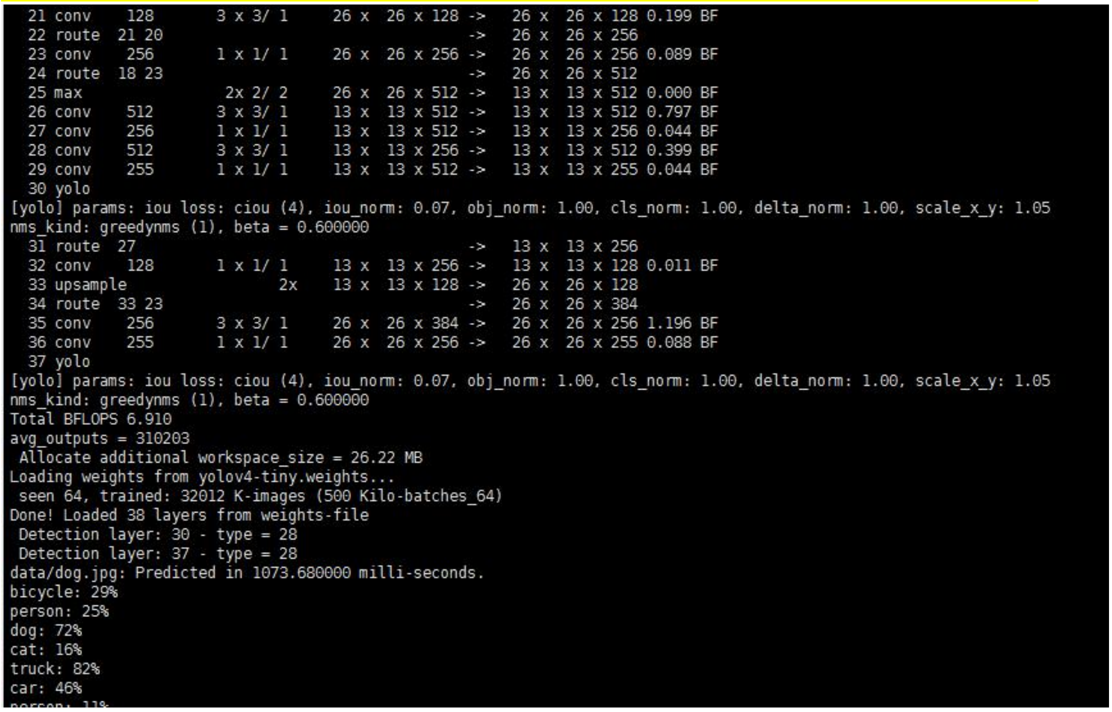
./darknet detector test cfg/coco.data cfg/yolov4-tiny.cfg yolov4-tiny.weights data/dog.jpg   # Full version

# Change the detection threshold

# By default, YOLO only displays detected objects with a confidence of .25 or higher. You can change this setting by passing the -thresh flag to the yolo command.

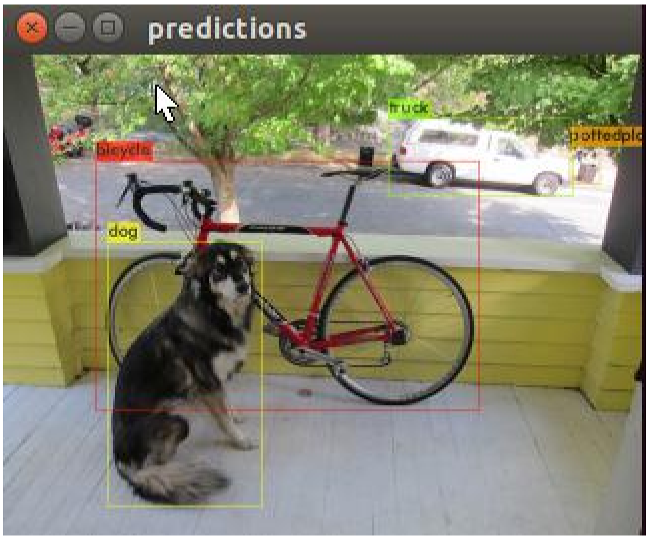
For example, to display all detections, you can set the threshold to 0.1.

./darknet detect cfg/yolov4-tiny.cfg yolov4-tiny.weights data/dog.jpg -thresh 0.1



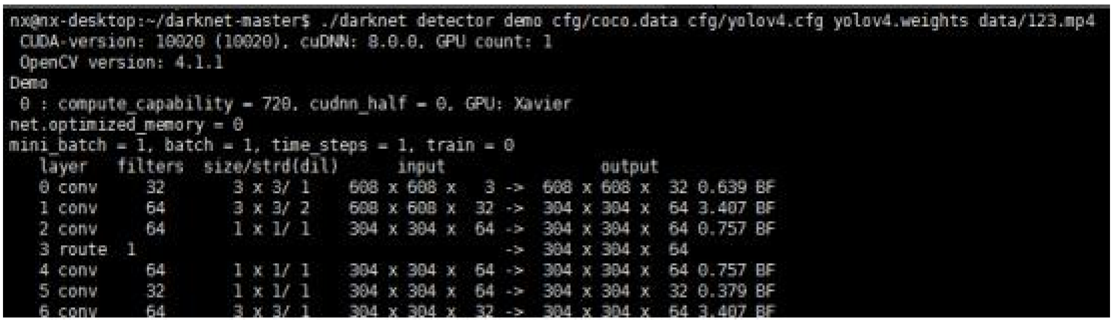
```
  21 conv    128       3 x 3/ 1     26 x  26 x 128 ->   26 x  26 x 128 0.199 BF
  22 route  21 20                                  ->   26 x  26 x 256
  23 conv    256       1 x 1/ 1     26 x  26 x 256 ->   26 x  26 x 256 0.089 BF
  24 route  18 23                                  ->   26 x  26 x 512
  25 max                2x 2/ 2     26 x  26 x 512 ->   13 x  13 x 512 0.000 BF
  26 conv    512       3 x 3/ 1     13 x  13 x 512 ->   13 x  13 x 512 0.797 BF
  27 conv    256       1 x 1/ 1     13 x  13 x 512 ->   13 x  13 x 256 0.044 BF
  28 conv    512       3 x 3/ 1     13 x  13 x 256 ->   13 x  13 x 512 0.399 BF
  29 conv    255       1 x 1/ 1     13 x  13 x 512 ->   13 x  13 x 255 0.044 BF
  30 yolo
[yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.05
nms_kind: greedynms (1), beta = 0.600000
  31 route  27                                     ->   13 x  13 x 256
  32 conv    128       1 x 1/ 1     13 x  13 x 256 ->   13 x  13 x 128 0.011 BF
  33 upsample                2x     13 x  13 x 128 ->   26 x  26 x 128
  34 route  33 23                                  ->   26 x  26 x 384
  35 conv    256       3 x 3/ 1     26 x  26 x 384 ->   26 x  26 x 256 1.196 BF
  36 conv    255       1 x 1/ 1     26 x  26 x 256 ->   26 x  26 x 255 0.088 BF
  37 yolo
[yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.05
nms_kind: greedynms (1), beta = 0.600000
Total BFLOPS 6.910
avg_outputs = 310203
 Allocate additional workspace_size = 26.22 MB
Loading weights from yolov4-tiny.weights...
 seen 64, trained: 32012 K-images (500 Kilo-batches_64)
Done! Loaded 38 layers from weights-file
 Detection layer: 30 - type = 28
 Detection layer: 37 - type = 28
data/dog.jpg: Predicted in 1073.680000 milli-seconds.
bicycle: 29%
person: 25%
dog: 72%
cat: 16%
truck: 82%
car: 46%
person: 11%
```

Result:



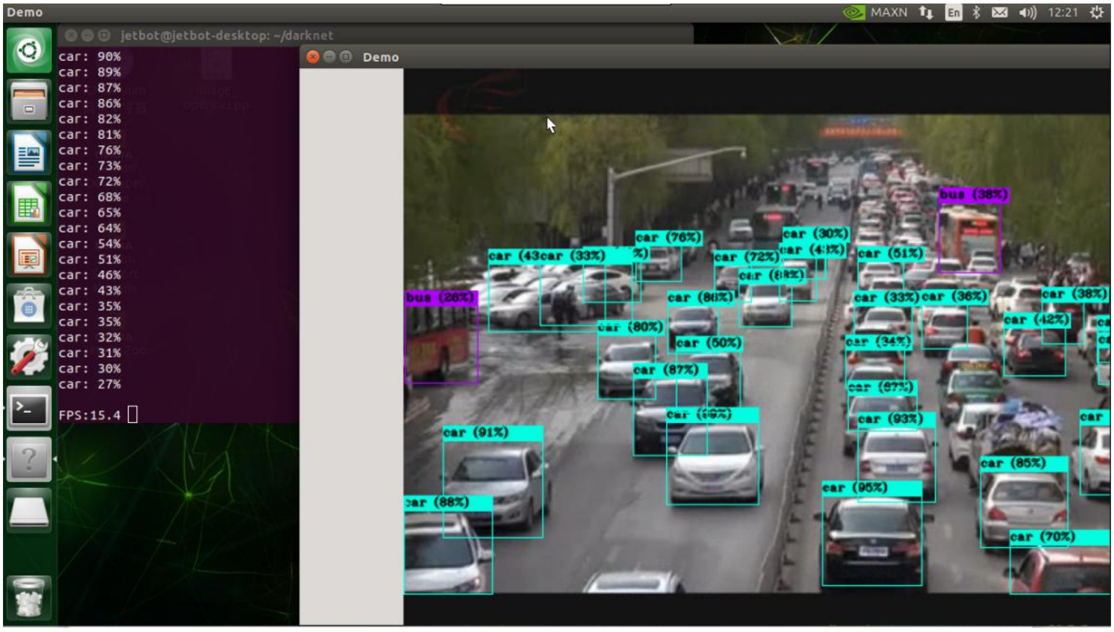
Yolov4-tiny video detection (the video file is not included in the data from github, and the user needs to upload the video file to be detected to the data folder)

./darknet detector demo cfg/coco.data cfg/yolov4-tiny.cfg yolov4-tiny.weights data/xxx.mp4



```
nx@nx-desktop:~/darknet-master$ ./darknet detector demo cfg/coco.data cfg/yolov4.cfg yolov4.weights data/123.mp4
 CUDA-version: 10020 (10020), cuDNN: 8.0.0, GPU count: 1
 OpenCV version: 4.1.1
Demo
 0 : compute_capability = 720, cudnn_half = 0, GPU: Xavier
net.optimized_memory = 0
mini_batch = 1, batch = 1, time_steps = 1, train = 0
   layer   filters  size/strd(dil)      input                output
   0 conv     32       3 x 3/ 1    608 x 608 x   3 ->  608 x 608 x  32 0.639 BF
   1 conv     64       3 x 3/ 2    608 x 608 x  32 ->  304 x 304 x  64 3.407 BF
   2 conv     64       1 x 1/ 1    304 x 304 x  64 ->  304 x 304 x  64 0.757 BF
   3 route  1                                    ->  304 x 304 x  64
   4 conv     64       1 x 1/ 1    304 x 304 x  64 ->  304 x 304 x  64 0.757 BF
   5 conv     32       1 x 1/ 1    304 x 304 x  64 ->  304 x 304 x  32 0.379 BF
   6 conv     64       3 x 3/ 1    304 x 304 x  32 ->  304 x 304 x  64 3.407 BF
```

yolov4-tiny, as shown below.



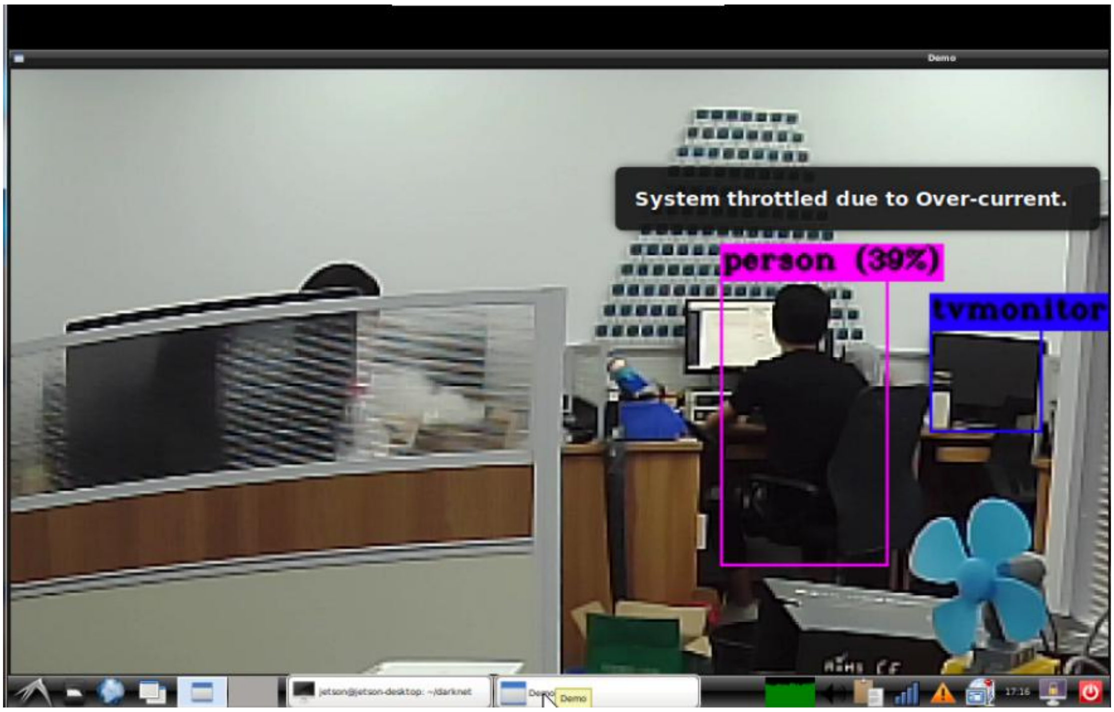
```
car: 90%
car: 89%
car: 87%
car: 86%
car: 82%
car: 81%
car: 76%
car: 73%
car: 72%
car: 68%
car: 65%
car: 64%
car: 54%
car: 51%
car: 46%
car: 43%
car: 35%
car: 35%
car: 32%
car: 31%
car: 30%
car: 27%

FPS:15.4
```

Yolov4-tiny camera real-time detection method:

```
./darknet detector demo cfg/coco.data cfg/yolov4-tiny.cfg yolov4-tiny.weights /dev/video1
```

Note:

The video device selects the number corresponding to the USB camera, and the number above is video1 with the USB camera selected.



System throttled due to Over-current.