

## 10、YoLoV4 环境搭建和摄像头实时检测

因为用的是 opencv4 所以使用 yolo3 编译可能回出错，所以这里我们采用 yolov4，同时 YOLO V4 无论在精度和速度上都较 YOLO V3 有了很大的提升，为在性能受限的嵌入式设备上部署检测程序提供了可能。闲话少说，开始安装。

### 1. 安装 CUDA, OpenCV, cuDNN

具体教程请看教程 1

### 2. 下载

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

注意:可能会出现 `error: RPC failed; curl 56 GnuTLS recv error (-54): Error in the pull function.` 这个原因是由于 git 默认缓存大小不足导致的，使用下面的命令增加缓存大小

```
git config --global http.postBuffer 524288000
```

如果还是不行可能是网速慢，配置 git 的最低速度和最低速度时间

```
git config --global http.lowSpeedLimit 0
```

```
git config --global http.lowSpeedTime 999999
```

如果还是不行以本人经历来说是网络问题，本人是在夜间所在网络使用人数少的时候成功的。

也可以通过到这个 <https://github.com/AlexeyAB/darknet.git> 下载 ZIP 文件到电脑，然后在移动 nx 里面解压。

### 3. 配置

```
cd darknet
```

```
sudo vim Makefile    #修改 Makefile
```

### 4. 将 Makefile 的前三行修改一下

```
GPU=1
```

```
CUDNN=1
```

```
OPENCV=1
```

### 5. 编译

在 darknet-master 路径下编译

```
make -j4
```

6. 下载权重文件，这里直接到.\5、常用的库和模型文件\yolo4 权重文件将权重文件 yolo4.weights 拷贝至 darknet 目录下

```

nx@nx-desktop: ~/darknet-master$ ls
3rdparty      darknet      json_mjpeg_streams.sh  scripts
backup        DarknetConfig.cmake.in LICENSE           src
build         darknet.py   Makefile          video_v2.sh
build.ps1     darknet_video.py net_cam_v3.sh      video_yolov3.sh
build.sh      data         obj               yolov4.weights
cfg           image_yolov2.sh predictions.jpg
cmake         image_yolov3.sh README.md
CMakeLists.txt include      results

```

## 7. 测试

图片的检测

`./darknet detect cfg/yolov4.cfg yolov4.weights data/dog.jpg` # 简写版

`./darknet detector test cfg/coco.data cfg/yolov4.cfg yolov4.weights data/dog.jpg` # 完整版

# 改变检测阈值

# 默认情况下, YOLO 仅显示检测到的置信度为.25 或更高的对象。您可以通过将 `-thresh <val>` 标志传递给 `yolo` 命令来更改此设置。

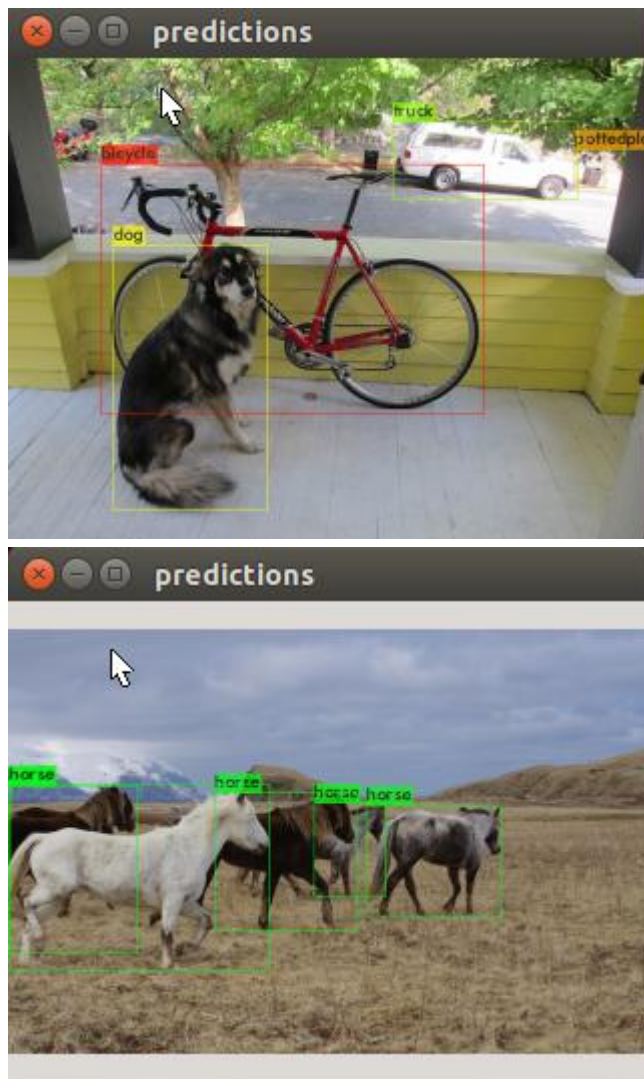
例如, 要显示所有检测, 您可以将阈值设置为 0.1:

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

```

nx@nx-desktop:~/darknet-master$ ./darknet detect cfg/yolov4.cfg yolov4.weights data/dog.jpg
CUDA-version: 10020 (10020), cudNN: 8.0.0, GPU count: 1
OpenCV version: 4.1.1
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
7 Shortcut Layer: 4, wt = 0, wn = 0, outputs: 304 x 304 x 64 0.006 BF
8 conv    64          1 x 1/ 1      304 x 304 x 64 -> 304 x 304 x 64 0.757 BF
9 route   8 2        ->          304 x 304 x 128
10 conv   64          1 x 1/ 1      304 x 304 x 128 -> 304 x 304 x 64 1.514 BF
11 conv   128         3 x 3/ 2      304 x 304 x 64 -> 152 x 152 x 128 3.407 BF
12 conv   64          1 x 1/ 1      152 x 152 x 128 -> 152 x 152 x 64 0.379 BF
13 route  11        ->          152 x 152 x 128
14 conv   64          1 x 1/ 1      152 x 152 x 128 -> 152 x 152 x 64 0.379 BF
15 conv   64          1 x 1/ 1      152 x 152 x 64 -> 152 x 152 x 64 0.189 BF
16 conv   64          3 x 3/ 1      152 x 152 x 64 -> 152 x 152 x 64 1.703 BF
17 Shortcut Layer: 14, wt = 0, wn = 0, outputs: 152 x 152 x 64 0.001 BF
18 conv   64          1 x 1/ 1      152 x 152 x 64 -> 152 x 152 x 64 0.189 BF
19 conv   64          3 x 3/ 1      152 x 152 x 64 -> 152 x 152 x 64 1.703 BF
20 Shortcut Layer: 17, wt = 0, wn = 0, outputs: 152 x 152 x 64 0.001 BF

```



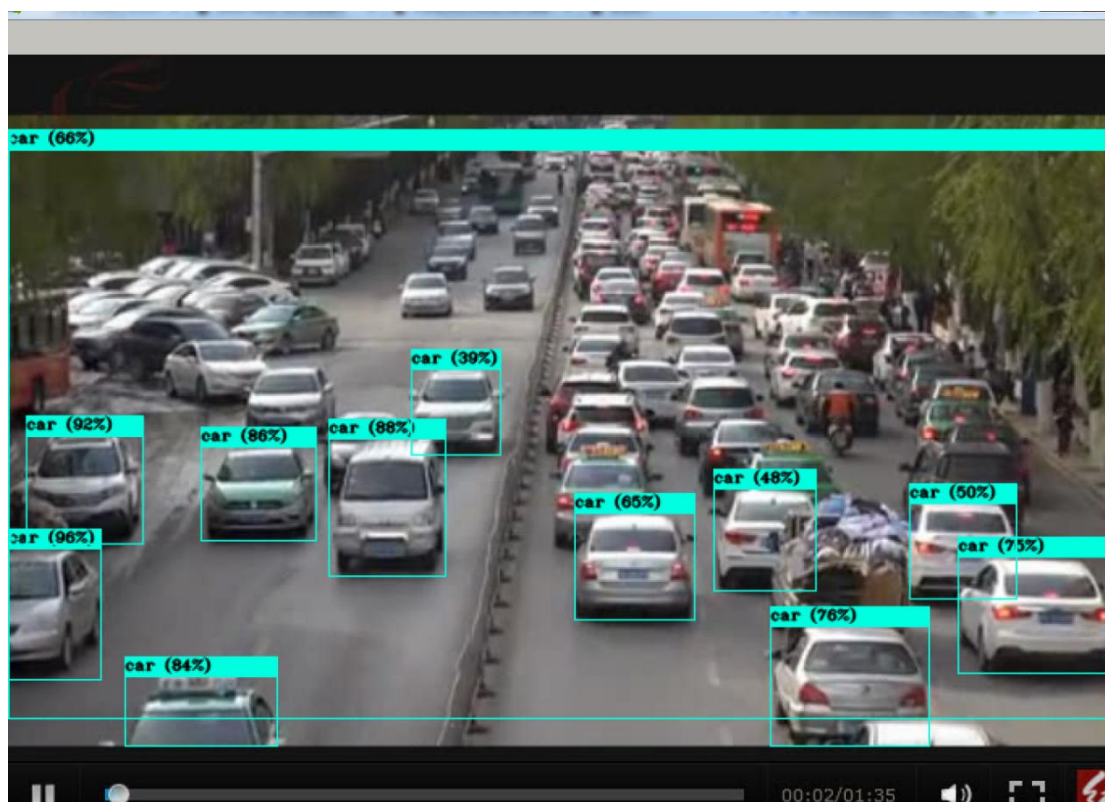
视频的检测

./darknet detector demo cfg/coco.data cfg/yolov4.cfg yolov4.weights data/123.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
  
```





摄像头实时检测方法:

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

注意 video 设备选择 USB 摄像头对应的编号

