海思3516移植YOLOV5步骤

# [Hi3516上yolov5的pytorch转onnx转caffe转wk详解](https://www.cnblogs.com/d442130165/p/16711189.html" \o "发布于 2022-09-20 15:33)

1. **环境配置**

VMware Workstation Pro 16.2.4

Ubuntu 18.04.6

Pytorch 1.10.0

Caffe-cpu 1.0

Python 3.6

Opencv 3.2.0

2. **下载并安装虚拟机**

下载地址：<https://customerconnect.vmware.com/cn/downloads/details?downloadGroup=WKST-1624-WIN&productId=1038&rPId=91434>

安装秘钥：

ZF3R0-FHED2-M80TY-8QYGC-NPKYF

3. **下载并安装Ubuntu18.04**

下载地址：<https://releases.ubuntu.com/18.04/ubuntu-18.04.6-desktop-amd64.iso>

4. **安装Pytorch1.10.0，注意ubuntu有默认Python为Python2，下面均使用Python3**

5.1修改pip3源

sudo apt update

sudo apt upgrade

sudo apt install vim

sudo apt install python3-pip

sudo pip3 install --upgrade pip

修改源

mkdir ~/.pip

vim ~/.pip/pip.conf

[global]

index-url = https://pypi.tuna.tsinghua.edu.cn/simple

或者

pip config set global.index-url https://pypi.mirrors.ustc.edu.cn/simple/

vim .config/pip/pip.conf 或 pip config list

可选安装pytorch，或者在yolov5的reqirements.txt一起安装

pip3 install torch==1.10.0+cpu torchvision==0.11.0+cpu torchaudio==0.10.0 -f https://download.pytorch.org/whl/torch\_stable.html

5. **下载YOLOv5-4.0，修改网络结构，进行训练，也可下载yolov5-6.0**

6.1下载Yolov5-4.0

下载地址：<https://codeload.github.com/ultralytics/yolov5/zip/refs/tags/v4.0>

6.2安装依赖文件

cd yolov5-4.0

pip3 install -r requirements.txt

6.3 添加数据集和配置文件

cd yolov5-4.0

vim data/roadsign\_voc.yaml

train: ../data/train.txt # train images (relative to 'path') 128 images

val: ../data/valid.txt # val images (relative to 'path') 128 images

test: # test images (optional)

# Classes

nc: 4 # number of classes

names: ['speedlimit', 'crosswalk', 'trafficlight', 'stop'] # class names

[记得这里要自己手动创建data和vaild.txt，然后修改其中的路径为绝对路径，两个txt里面也是绝对路径去寻找照片，标红部分指需要修改路径]

6.4 修改网络配置

cd yolov5-4.0

vim models/yolov5s.yaml

[IMG_256](javascript:void(0);)

# parameters

nc: 80 # number of classes

depth\_multiple: 0.33 # model depth multiple

width\_multiple: 0.50 # layer channel multiple

# anchors

anchors:

- [10,13, 16,30, 33,23] # P3/8

- [30,61, 62,45, 59,119] # P4/16

- [116,90, 156,198, 373,326] # P5/32

# YOLOv5 backbone

backbone:

# [from, number, module, args]

[

#[-1, 1, Focus, [64, 3]], # 0-P1/2

[-1, 1, Conv, [64, 3,2]], # 0-P1/2

[-1, 1, Conv, [128, 3, 2]], # 1-P2/4

[-1, 3, C3, [128]],

[-1, 1, Conv, [256, 3, 2]], # 3-P3/8

[-1, 9, C3, [256]],

[-1, 1, Conv, [512, 3, 2]], # 5-P4/16

[-1, 9, C3, [512]],

[-1, 1, Conv, [1024, 3, 2]], # 7-P5/32

[-1, 1, SPP, [1024, [5, 9, 13]]],

[-1, 3, C3, [1024, False]], # 9

]

# YOLOv5 headhead:

[[-1, 1, Conv, [512, 1, 1]],

#[-1, 1, nn.Upsample, [None, 2, 'nearest']],

[-1, 1, nn.ConvTranspose2d, [256, 256, 2,2]],

[[-1, 6], 1, Concat, [1]], # cat backbone P4

[-1, 3, C3, [512, False]], # 13

[-1, 1, Conv, [256, 1, 1]],

#[-1, 1, nn.Upsample, [None, 2, 'nearest']],

[-1, 1, nn.ConvTranspose2d, [128, 128, 2,2]],

[[-1, 4], 1, Concat, [1]], # cat backbone P3

[-1, 3, C3, [256, False]], # 17 (P3/8-small)

[-1, 1, Conv, [256, 3, 2]],

[[-1, 14], 1, Concat, [1]], # cat head P4

[-1, 3, C3, [512, False]], # 20 (P4/16-medium)

[-1, 1, Conv, [512, 3, 2]],

[[-1, 10], 1, Concat, [1]], # cat head P5

[-1, 3, C3, [1024, False]], # 23 (P5/32-large)

[[17, 20, 23], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5)

]

[IMG_257](javascript:void(0);)

6.5 修改网络结构

cd yolov5-4.0

vim models/common.py

[IMG_258](javascript:void(0);)

class Conv(nn.Module):

# Standard convolution

def \_\_init\_\_(self, c1, c2, k=1, s=1, p=None, g=1, act=True): # ch\_in, ch\_out, kernel, stride, padding, groups

super(Conv, self).\_\_init\_\_()

self.conv = nn.Conv2d(c1, c2, k, s, autopad(k, p), groups=g, bias=False)

self.bn = nn.BatchNorm2d(c2)

# self.act = nn.SiLU() if act is True else (act if isinstance(act, nn.Module) else nn.Identity())

self.act = nn.ReLU() if act is True else (act if isinstance(act, nn.Module) else nn.Identity())

def forward(self, x):

return self.act(self.bn(self.conv(x)))

def fuseforward(self, x):

return self.act(self.conv(x))

class BottleneckCSP(nn.Module):

# CSP Bottleneck https://github.com/WongKinYiu/CrossStagePartialNetworks

def \_\_init\_\_(self, c1, c2, n=1, shortcut=True, g=1, e=0.5): # ch\_in, ch\_out, number, shortcut, groups, expansion

super(BottleneckCSP, self).\_\_init\_\_()

c\_ = int(c2 \* e) # hidden channels

self.cv1 = Conv(c1, c\_, 1, 1)

self.cv2 = nn.Conv2d(c1, c\_, 1, 1, bias=False)

self.cv3 = nn.Conv2d(c\_, c\_, 1, 1, bias=False)

self.cv4 = Conv(2 \* c\_, c2, 1, 1)

self.bn = nn.BatchNorm2d(2 \* c\_) # applied to cat(cv2, cv3)

# self.act = nn.LeakyReLU(0.1, inplace=True)

self.act = nn.ReLU()

self.m = nn.Sequential(\*[Bottleneck(c\_, c\_, shortcut, g, e=1.0) for \_ in range(n)])

def forward(self, x):

y1 = self.cv3(self.m(self.cv1(x)))

y2 = self.cv2(x)

return self.cv4(self.act(self.bn(torch.cat((y1, y2), dim=1))))

class SPP(nn.Module):

# Spatial pyramid pooling layer used in YOLOv3-SPP

def \_\_init\_\_(self, c1, c2, k=(5, 9, 13)):

super(SPP, self).\_\_init\_\_()

c\_ = c1 // 2 # hidden channels

self.cv1 = Conv(c1, c\_, 1, 1)

self.cv2 = Conv(c\_ \* (len(k) + 1), c2, 1, 1)

# self.m = nn.ModuleList([nn.MaxPool2d(kernel\_size=x, stride=1, padding=x // 2) for x in k])

self.m = nn.ModuleList([nn.MaxPool2d(kernel\_size=x, stride=1, padding=x // 2, ceil\_mode=True) for x in k])

def forward(self, x):

x = self.cv1(x)

return self.cv2(torch.cat([x] + [m(x) for m in self.m], 1))

[IMG_259](javascript:void(0);)

6.6 训练模型

cd yolov5-4.0/weights

wget [https://github.com/ultralytics/yolov5/releases/download/v4.0/yolov5s.pt](https://github.com/ultralytics/yolov5/releases/download/v3.1/yolov5s.pt)

python3 train.py --img 640 --batch 16 --epochs 1 --data data/roadsign\_voc.yaml --cfg models/yolov5s.yaml --weights weights/yolov5s.pt --noautoanchor

这里训练出现了两个问题：  
1.没有手动创建vaild.txt和train.txt，没有创建labels[这个文件夹也要手动创建]下的bus.txt和person.txt的标签文件（roadsign\_voc.yaml）

1. 在虚拟机下yolov5.pt文件显示为 ’yolov5.pt’（有双引号说明有奇怪的空格），正常yolovvv5.pt没有任何显示才是正常的，如果没有找到.pt会自动到官网下载，不仅很慢而且因为版本非常新导致要使用到SPPF层，而我们的commn.py只有SPP层会导致报错
2. 必须要自己手动添加之后手动训练之后产生的exp..才可以继续进行执行，其他不是自己执行指令生成的后面会报各种错

训练成功的模型文件路径：runs/train/exp12(这个看训练结果提示信息来)/weights



Last.pt best.pt 这张图是我已经进行了onnx转化的

6. **导出onnx模型**

pip3 install onnx==1.8.1

pip3 install onnx-simplifier

cd yolov5-4.0/

vim models/export.py

model.model[-1].export = True

torch.onnx.export(model, img, f, verbose=False, opset\_version=10, input\_names=['images'], output\_names=['classes', 'boxes'] if y is None else ['output'])

python3 models/export.py --weights runs/train/exp/weights/last.pt

python3 -m onnxsim runs/train/exp/weights/last.onnx runs/train/exp/weights/yolov5s\_sim.onnx

转化成功的onnx文件路径：runs/train/exp12/weights

结果如下：



7. **下载并安装yolov5\_caffe，编译生成caffe和pycaffe**

7.1下载地址：

<https://codeload.github.com/Wulingtian/yolov5_caffe/zip/refs/heads/master>

7.2安装依赖包

sudo apt-get install libprotobuf-dev libleveldb-dev libsnappy-dev libopencv-dev libhdf5-serial-dev protobuf-compiler

sudo apt-get install --no-install-recommends libboost-all-dev

sudo apt-get install python3-dev

sudo apt-get install libatlas-base-dev

sudo apt-get install libgflags-dev libgoogle-glog-dev liblmdb-dev

sudo apt-get install python3-opencv

7.3修改安装配置文件，使用CPU版本进行编译安装caffe

cd yolov5\_caffe

**vim Makefile.config**

[IMG_260](javascript:void(0);)

# USE\_CUDNN := 1

CPU\_ONLY := 1

OPENCV\_VERSION := 3

BLAS := atlas

# CUDA\_DIR := /usr

# CUDA\_ARCH := -gencode arch=compute\_61,code=compute\_61

# ANACONDA\_HOME := $(HOME)/anaconda

# PYTHON\_INCLUDE := $(ANACONDA\_HOME)/include \

# $(ANACONDA\_HOME)/include/python3.6m \

# $(ANACONDA\_HOME)/lib/python3.6/site-packages/numpy/core/include

PYTHON\_LIBRARIES := boost\_python3 python3.6m

PYTHON\_INCLUDE := /usr/include/python3.6m \

/usr/lib/python3.6/dist-packages/numpy/core/include

#PYTHON\_LIBRARIES := boost\_python-py35

# PYTHON\_INCLUDE := /usr/include/python3.5m \

# /usr/lib/python3.5/dist-packages/numpy/core/include

PYTHON\_LIB := /usr/lib

# PYTHON\_LIB := $(ANACONDA\_HOME)/lib

WITH\_PYTHON\_LAYER := 1

INCLUDE\_DIRS := $(PYTHON\_INCLUDE) /usr/local/include /usr/include/hdf5/serial

LIBRARY\_DIRS := $(PYTHON\_LIB) /usr/local/lib /usr/lib /usr/lib/x86\_64-linux-gnu /usr/lib/x86\_64-linux-gnu/hdf5/serial

BUILD\_DIR := build

DISTRIBUTE\_DIR := distribute

# TEST\_GPUID := 0

Q ?= @

[IMG_261](javascript:void(0);)

**vim Makefile**

LIBRARIES += glog gflags protobuf boost\_system boost\_filesystem m hdf5\_serial\_hl hdf5\_serial

验证是否安装成功

make clean

make all -j8

make test -j8

make runtest -j8

7.4安装pycaffe

cd yolov5\_caffe/python

for req in (catrequirements.txt);dopipinstall(catrequirements.txt);dopipinstallreq; done

或者

sudo apt-get install python-numpy python-scipy python-matplotlib python-sklearn python-skimage python-h5py python-protobuf python-leveldb python-networkx python-nose python-pandas python-gflags Cython ipython

sudo apt-get install protobuf-c-compiler protobuf-compiler

cd yolov5\_caffe/

make pycaffe -j8

vim ~/.bashrc

export PYTHONPATH=.../yolov5\_caffe/python:$PYTHONPATH

source ~/.bashrc

验证是否安装成功

python3

import caffe

8. **下载yolov5\_onnx2caff，把onnx模型转换为caffe模型**

下载地址：

<https://codeload.github.com/Hiwyl/yolov5_onnx2caffe/zip/refs/heads/master>

cd yolov5\_onnx2caffe/

vim convertCaffe.py

onnx\_path = "./weights/yolov5s\_sim.onnx"

prototxt\_path = "./weights/yolov5s\_sim.prototxt"

caffemodel\_path = "./weights/yolov5s\_sim.caffemodel"

convertToCaffe(graph, prototxt\_path, caffemodel\_path)

 这里作者漏了一句执行语句将onnx进行转化，我们需要到yolov5\_onnx2caffe/文件夹下自己手动创建weights文件夹将前面转化好的onnx文件（yolov5s\_sim.onnx）拷贝过来，然后到上一目录执行：python3 convertCaffe.py

进行模型转化，转化好的结果在yolov5\_onnx2caffe/weights路径下

结果如下：



9. **验证转换的caffe模型**

cd yolov5\_caffe/tools

设置输入参数和模型路径

vim caffe\_yolov5s.cpp

[IMG_262](javascript:void(0);)

#define INPUT\_W 640#define INPUT\_H 640#define IsPadding 1#define NUM\_CLASS 4#define NMS\_THRESH 0.6#define CONF\_THRESH 0.3

std::string prototxt\_path = "./weights/yolov5s\_sim.prototxt";

std::string caffemodel\_path = "./weithgts/yolov5s\_sim.caffemodel";

std::string pic\_path = "./weights/road580.png";

[IMG_263](javascript:void(0);)

编译caffe\_yolov5s

cd yolov5\_caffe/

sudo apt install cmake

make

运行caffe\_yolov5s

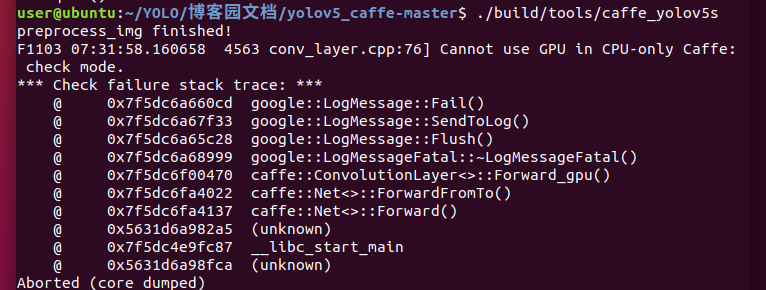
cp ../yolov5\_onnx2caffe/weights/yolov5s\_sim.caffemodel weights/

cp ../yolov5\_onnx2caffe/weights/yolov5s\_sim.prototxt weights/

./build/tools/caffe\_yolov5s

会遇到3个问题：

1. 这里原文件夹是没有 road580.png这张照片的，需要自己添加，并且将分辨率设置成640\*640，并且这里也要手动创建weights文件夹
2. 这里会报一个错：



可以看到根据前面的修改和操作我们的caffe模型确实是CPU版本的，但是却在使用GPU进行操作，问题出在caffe\_yolov5s.cpp[tools/下]这个文件里，他的源码文档没有给出一个地方的修改就是将原本的

Caffe::set\_mode(Caffe::GPU);这句应该改成Caffe::set\_mode(Caffe::CPU);Caffe::set\_mode(Caffe::CPU);使用CPU

修改完之后我们发现还是同一个报错，仔细看看语句，发现我们执行的是可执行文件，而我们修改了.cpp，所以我们需要重新编译然后再执行，退出到上一文件夹：

编译caffe\_yolov5s

cd yolov5\_caffe/

make

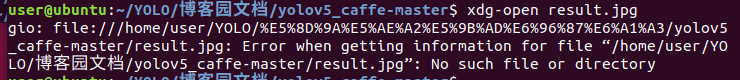
下面是重新编译生成可执行文件成功：



查看检测结果

xdg-open result.jpg

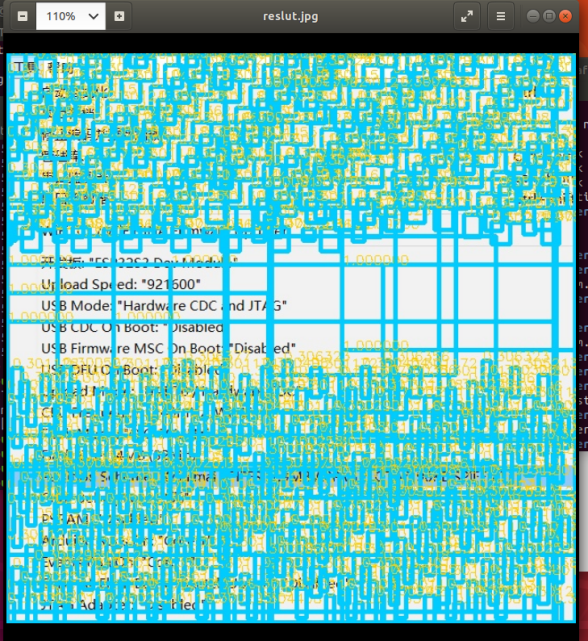
注意这里会报一个错：



其实是源文件不知道为啥（可能是自己改的），里面的这一句输出语句：cv::imwrite("result.jpg", showImage);

变成了cv::imwrite("reslut.jpg", showImage);

下面是检测结果：



10. **安装nnie mapper，把caffe模型转换为wk模型**

10.1 在hi3516dv300中提供的资料中找到nnie\_mapper\_12，安装参考文档HiSVP 开发指南的 Linux 版 NNIE mapper 安装

nnie\_mapper\_12位置在SVP\_PC\HiSVP\_PC\_V1.2.2.2\tools\nnie\linux\mapper中

使用readelf查看nnie\_mapper\_12的依赖库，需要protobuf3.6.1和opencv3.4

readelf -d nnie\_mapper\_12

10.2 安装opencv3.4.2

下载地址：<https://github.com/opencv/opencv/archive/3.4.2.zip>

安装依赖

sudo apt-get install build-essential

sudo apt-get install cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev

sudo apt-get install python-dev python-numpy libtbb2 libtbb-dev libjpeg-dev libtiff-dev libjasper-dev libdc1394-22-dev

安装opencv

mkdir yolov5\_caffe2nnie

cd yolov5\_caffe2nnie

mkdir 3rd

cd 3rd

tar -xvzf opencv-3.4.2.tar.gz

cd opencv-3.4.2

mkdir build && cd build

cmake -D CMAKE\_BUILD\_TYPE=RELEASE -D CMAKE\_INSTALL\_PREFIX=.../yolov5\_caffe2nnie ../

make -j8

make install

添加环境路径

vim ~/.bashrc

export PATH=.../yolov5\_caffe2nnie/bin:$PATH

export LD\_LIBRARY\_PATH=.../yolov5\_caffe2nnie/lib:$LD\_LIBRARY\_PATH

export PKG\_CONFIG\_PATH=.../yolov5\_caffe2nnie/lib/pkgconfig

source ~/.bashrc

验证opencv是否安装成功

运行opencv\_version显示3.4.2

编译opencv程序

sudo apt-get install libcanberra-gtk-module

cd opencv-3.4.2/samples/cpp/

 gcc `pkg-config --cflags opencv` -o facedetect facedetect.cpp `pkg-config --libs opencv` -lstdc++

./facedetect ../data/lena.jpg

10.3 安装protobuf3.6.1

protobuf3.6.1编译需要用gcc4.8，Ubuntu18.04自带gcc7.5，需要安装多版本gcc

sudo apt update

sudo apt install build-essential

sudo apt install software-properties-common

sudo add-apt-repository ppa:ubuntu-toolchain-r/test

sudo apt install gcc-4.8 g++-4.8 gcc-7 g++-7 gcc-8 g++-8 gcc-9 g++-9

sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.8 48 --slave /usr/bin/g++ g++ /usr/bin/g++-4.8

sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-7 70 --slave /usr/bin/g++ g++ /usr/bin/g++-7

sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-8 80 --slave /usr/bin/g++ g++ /usr/bin/g++-8

sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-9 90 --slave /usr/bin/g++ g++ /usr/bin/g++-9

切换到gcc4.8

sudo update-alternatives --config gcc

gcc -v

下载protobuf3.6.1：

<https://github.com/google/protobuf/releases/download/v3.6.1/protobuf-all-3.6.1.tar.gz>

tar -xvf protobuf.3.6.1

cd protobuf3.6.1

./configure -prefix=.../yolov5\_caffe2nnie

make

make check

make install

验证protobuf.3.6.1

protoc --version

显示libprotoc 3.6.1

切换到gcc7.5

sudo update-alternatives --config gcc

gcc -v

10.4 安装nnie\_mapper\_12

复制nnie\_mapper\_12到.../yolov5\_caffe2nnie/bin

验证nnie\_mapper\_12

复制SVP\_PC\HiSVP\_PC\_V1.2.2.2\software\data到.../yolov5\_caffe2nnie/

cd .../yolov5\_caffe2nnie/data

运行nnie\_mapper\_12 classification/alexnet/alexnet\_no\_group\_inst.cfg

显示下列信息，说明安装成功

[IMG_264](javascript:void(0);)

Mapper Version 1.2.2.1\_B030 (NNIE\_1.2) 19090610466402

begin net parsing....

end net parsing

begin prev optimizing....

end prev optimizing....

begin net quantalizing(CPU)....

end quantalizing

begin optimizing....

end optimizing

begin NNIE[0] mem allocation....

end NNIE[0] memory allocating

begin NNIE[0] instruction generating....

end NNIE[0] instruction generating

begin parameter compressing....

end parameter compressing

begin compress index generating....

end compress index generating

begin binary code generating....

end binary code generating

begin quant files writing....

end quant files writing

[IMG_265](javascript:void(0);)

10.5 修改yolov5的caffe网络结构，并转换为wk模型

复制onnx转换好的caffe模型到.../yolov5\_caffe2nnie/

cp ../yolov5\_onnx2caffe/weights/yolov5s\_sim.caffemodel ./weights/

cp ../yolov5\_onnx2caffe/weights/yolov5s\_sim.prototxt ./weights/

修改yolov5的caffe网络结构的三个输出头部

vim yolov5s\_sim.prototxt

删除三个Permute

修改三个Reshape为下面格式

[IMG_266](javascript:void(0);)

layer {

name: "Reshape\_151"

type: "Reshape"

bottom: "265"

top: "277"

reshape\_param {

shape {

dim: 0

dim: 3

dim: 9

dim: 6400

}

}

}

layer {

name: "Reshape\_165"

type: "Reshape"

bottom: "279"

top: "291"

reshape\_param {

shape {

dim: 0

dim: 3

dim: 9

dim: 1600

}

}

}

layer {

name: "Reshape\_179"

type: "Reshape"

bottom: "293"

top: "305"

reshape\_param {

shape {

dim: 0

dim: 3

dim: 9

dim: 400

}

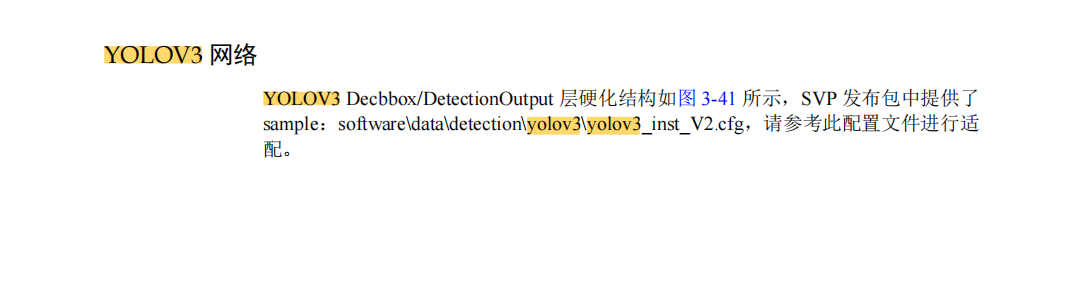
}

}

[IMG_267](javascript:void(0);)

编写转换配置文件，参考yolov3的配置文件，具体参数参考HiSVP 开发指南中的nnie\_mapperde的配置文件说明

 这里需要手动创建一个yolov5s\_sim.cfg文件，其实在SVP开发指南也可以找到参考：E:\BaiduNetdiskDownload\Hi3516CV500R001C02SPC020(齐全SDK)\Hi3516CV500R001C02SPC020\SVP\_PC\SVP\_PC\HiSVP\_PC\_V1.2.2.2\HiSVP\_PC\_V1.2.2.2\HiSVP\_PC\_V1.2.2.2\software\data\detection\yolov3



vim weights/yolov5s\_sim.cfg

[IMG_268](javascript:void(0);)

[prototxt\_file] ./weights/yolov5s\_sim.prototxt

[caffemodel\_file] ./weights/yolov5s\_sim.caffemodel

[batch\_num] 1

[net\_type] 0

[sparse\_rate] 0

[compile\_mode] 1

[is\_simulation] 0

[log\_level] 2

[instruction\_name] ./weights/yolov5s\_sim

[RGB\_order] BGR

[data\_scale] 0.0039062

[internal\_stride] 16

[image\_list] ../../data/valid.txt

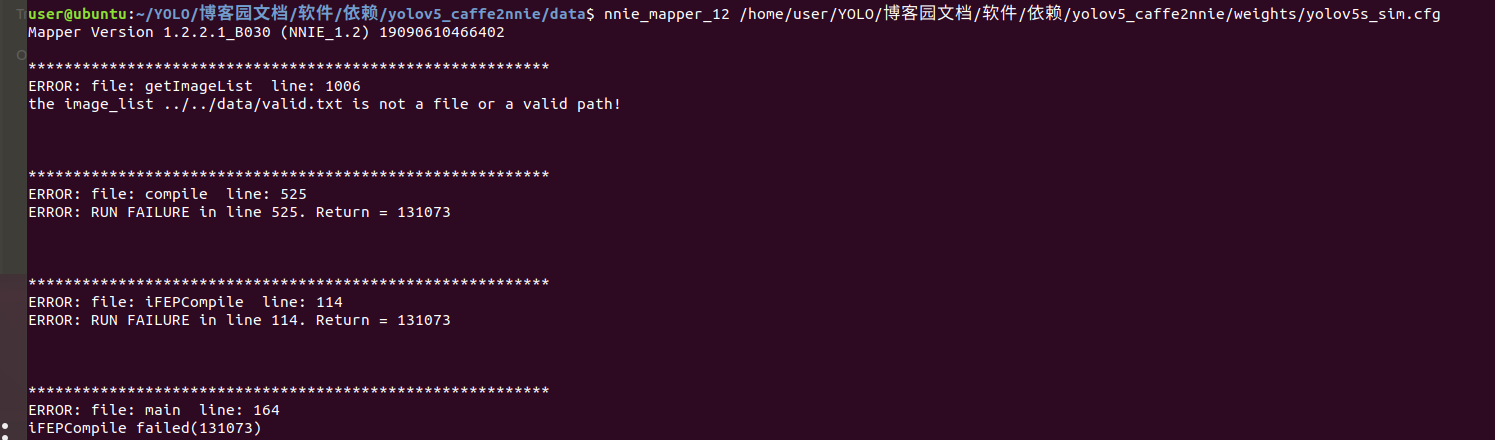
[image\_type] 1

[mean\_file] null

[norm\_type] 3

其实在弄完上面这一步就可以着手开始.wk的模型转化了，这时候我们会碰到问题：

1.路径问题



其实都是路径问题，不知道为啥用相对路径找不到文件，我的建议是将所有相对路径全部换成绝对路径就可以了

2.还有一个就是需要根据上面的配置文件创建一个valid.txt文件（原本没有），用来指向照片路径，并且最好将验证的原始照片再创建一个文件夹进行放置

[IMG_269](javascript:void(0);)

11. **下载YOLOv5-6.0，修改网络结构，进行训练**

11.1下载Yolov5-6.0

下载地址：<https://codeload.github.com/ultralytics/yolov5/zip/refs/tags/v4.0>

11.2安装依赖文件

cd yolov5-6.0

pip3 install -r requirements.txt

11.3 添加数据集和配置文件

cd yolov5-6.0

vim data/roadsign\_voc.yaml

train: ../data/train.txt  # train images (relative to 'path') 128 images

val: ../data/valid.txt  # val images (relative to 'path') 128 images

test:  # test images (optional)

# Classes

nc: 4  # number of classes

names: ['speedlimit', 'crosswalk', 'trafficlight', 'stop']   # class names

11.4 修改网络配置

cd yolov5-6.0

vim models/yolov5s.yaml

[IMG_270](javascript:void(0);)

# YOLOv5 v6.0 backbone

backbone:

# [from, number, module, args]

[[-1, 1, Conv, [64, 6, 2, 2]], # 0-P1/2

[-1, 1, Conv, [128, 3, 2]], # 1-P2/4

[-1, 3, C3, [128]],

[-1, 1, Conv, [256, 3, 2]], # 3-P3/8

[-1, 6, C3, [256]],

[-1, 1, Conv, [512, 3, 2]], # 5-P4/16

[-1, 9, C3, [512]],

[-1, 1, Conv, [1024, 3, 2]], # 7-P5/32

[-1, 3, C3, [1024]],

[-1, 1, SPPF, [1024, 5]], # 9

]

# YOLOv5 v6.0 headhead:

[[-1, 1, Conv, [512, 1, 1]],

#[-1, 1, nn.Upsample, [None, 2, 'nearest']],

[-1, 1, nn.ConvTranspose2d, [256, 256, 2, 2]],

[[-1, 6], 1, Concat, [1]], # cat backbone P4

[-1, 3, C3, [512, False]], # 13

[-1, 1, Conv, [256, 1, 1]],

#[-1, 1, nn.Upsample, [None, 2, 'nearest']],

[-1, 1, nn.ConvTranspose2d, [128, 128, 2, 2]],

[[-1, 4], 1, Concat, [1]], # cat backbone P3

[-1, 3, C3, [256, False]], # 17 (P3/8-small)

[-1, 1, Conv, [256, 3, 2]],

[[-1, 14], 1, Concat, [1]], # cat head P4

[-1, 3, C3, [512, False]], # 20 (P4/16-medium)

[-1, 1, Conv, [512, 3, 2]],

[[-1, 10], 1, Concat, [1]], # cat head P5

[-1, 3, C3, [1024, False]], # 23 (P5/32-large)

[[17, 20, 23], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5)

]

[IMG_271](javascript:void(0);)

6.5 修改网络结构

cd yolov5-6.0

vim models/common.py

[IMG_272](javascript:void(0);)

class Conv(nn.Module):

# Standard convolution

def \_\_init\_\_(self, c1, c2, k=1, s=1, p=None, g=1, act=True): # ch\_in, ch\_out, kernel, stride, padding, groups

super().\_\_init\_\_()

self.conv = nn.Conv2d(c1, c2, k, s, autopad(k, p), groups=g, bias=False)

self.bn = nn.BatchNorm2d(c2)

#self.act = nn.SiLU() if act is True else (act if isinstance(act, nn.Module) else nn.Identity())

self.act = nn.ReLU() if act is True else (act if isinstance(act, nn.Module) else nn.Identity())

def forward(self, x):

return self.act(self.bn(self.conv(x)))

def forward\_fuse(self, x):

return self.act(self.conv(x))

lass BottleneckCSP(nn.Module):

# CSP Bottleneck https://github.com/WongKinYiu/CrossStagePartialNetworks

def \_\_init\_\_(self, c1, c2, n=1, shortcut=True, g=1, e=0.5): # ch\_in, ch\_out, number, shortcut, groups, expansion

super().\_\_init\_\_()

c\_ = int(c2 \* e) # hidden channels

self.cv1 = Conv(c1, c\_, 1, 1)

self.cv2 = nn.Conv2d(c1, c\_, 1, 1, bias=False)

self.cv3 = nn.Conv2d(c\_, c\_, 1, 1, bias=False)

self.cv4 = Conv(2 \* c\_, c2, 1, 1)

self.bn = nn.BatchNorm2d(2 \* c\_) # applied to cat(cv2, cv3)

#self.act = nn.LeakyReLU(0.1, inplace=True)

self.act = nn.ReLU()

self.m = nn.Sequential(\*[Bottleneck(c\_, c\_, shortcut, g, e=1.0) for \_ in range(n)])

def forward(self, x):

y1 = self.cv3(self.m(self.cv1(x)))

y2 = self.cv2(x)

return self.cv4(self.act(self.bn(torch.cat((y1, y2), dim=1))))

class SPPF(nn.Module):

# Spatial Pyramid Pooling - Fast (SPPF) layer for YOLOv5 by Glenn Jocher

def \_\_init\_\_(self, c1, c2, k=5): # equivalent to SPP(k=(5, 9, 13))

super().\_\_init\_\_()

c\_ = c1 // 2 # hidden channels

self.cv1 = Conv(c1, c\_, 1, 1)

self.cv2 = Conv(c\_ \* 4, c2, 1, 1)

#self.m = nn.MaxPool2d(kernel\_size=k, stride=1, padding=k // 2)

self.m = nn.MaxPool2d(kernel\_size=k, stride=1, padding=k // 2, ceil\_mode=True)

def forward(self, x):

x = self.cv1(x)

with warnings.catch\_warnings():

warnings.simplefilter('ignore') # suppress torch 1.9.0 max\_pool2d() warning

y1 = self.m(x)

y2 = self.m(y1)

return self.cv2(torch.cat([x, y1, y2, self.m(y2)], 1))

[IMG_273](javascript:void(0);)

11.6 训练模型

cd yolov5-6.0/weights

wget [https://github.com/ultralytics/yolov5/releases/download/v6.0/yolov5s.pt](https://github.com/ultralytics/yolov5/releases/download/v3.1/yolov5s.pt)

python3 train.py --img 640 --batch 16 --epochs 1 --data data/roadsign\_voc.yaml --cfg models/yolov5s.yaml --weights weights/yolov5s.pt --noautoanchor

基本上照着做就有了，但是有些参数要修改

参考文档：

[Hi3516上yolov5的pytorch转onnx转caffe转wk详解 - 盛夏夜 - 博客园](https://www.cnblogs.com/d442130165/p/16711189.html)

下面是各个步骤的提示成功信息：

**1..模型训练.pt文件**

user@ubuntu:~/YOLO/博客园文档/yolov5-4.0$ python3 train.py --img 640 --batch 16 --epochs 1 --data data/roadsign\_voc.yaml --cfg models/yolov5s.yaml --weights weights/yolov5s.pt --noautoanchor

Using torch 1.10.0+cpu CPU

Namespace(adam=False, batch\_size=16, bucket='', cache\_images=False, cfg='models/yolov5s.yaml', data='data/roadsign\_voc.yaml', device='', epochs=1, evolve=False, exist\_ok=False, global\_rank=-1, hyp='data/hyp.scratch.yaml', image\_weights=False, img\_size=[640, 640], local\_rank=-1, log\_artifacts=False, log\_imgs=16, multi\_scale=False, name='exp', noautoanchor=True, nosave=False, notest=False, project='runs/train', quad=False, rect=False, resume=False, save\_dir='runs/train/exp13', single\_cls=False, sync\_bn=False, total\_batch\_size=16, weights='weights/yolov5s.pt', workers=8, world\_size=1)

Start Tensorboard with "tensorboard --logdir runs/train", view at http://localhost:6006/

Hyperparameters {'lr0': 0.01, 'lrf': 0.2, 'momentum': 0.937, 'weight\_decay': 0.0005, 'warmup\_epochs': 3.0, 'warmup\_momentum': 0.8, 'warmup\_bias\_lr': 0.1, 'box': 0.05, 'cls': 0.5, 'cls\_pw': 1.0, 'obj': 1.0, 'obj\_pw': 1.0, 'iou\_t': 0.2, 'anchor\_t': 4.0, 'fl\_gamma': 0.0, 'hsv\_h': 0.015, 'hsv\_s': 0.7, 'hsv\_v': 0.4, 'degrees': 0.0, 'translate': 0.1, 'scale': 0.5, 'shear': 0.0, 'perspective': 0.0, 'flipud': 0.0, 'fliplr': 0.5, 'mosaic': 1.0, 'mixup': 0.0}

Overriding model.yaml nc=80 with nc=2

from n params module arguments

0 -1 1 928 models.common.Conv [3, 32, 3, 2]

1 -1 1 18560 models.common.Conv [32, 64, 3, 2]

2 -1 1 18816 models.common.C3 [64, 64, 1]

3 -1 1 73984 models.common.Conv [64, 128, 3, 2]

4 -1 1 156928 models.common.C3 [128, 128, 3]

5 -1 1 295424 models.common.Conv [128, 256, 3, 2]

6 -1 1 625152 models.common.C3 [256, 256, 3]

7 -1 1 1180672 models.common.Conv [256, 512, 3, 2]

8 -1 1 656896 models.common.SPP [512, 512, [5, 9, 13]]

9 -1 1 1182720 models.common.C3 [512, 512, 1, False]

10 -1 1 131584 models.common.Conv [512, 256, 1, 1]

11 -1 1 262400 torch.nn.modules.conv.ConvTranspose2d [256, 256, 2, 2]

12 [-1, 6] 1 0 models.common.Concat [1]

13 -1 1 361984 models.common.C3 [512, 256, 1, False]

14 -1 1 33024 models.common.Conv [256, 128, 1, 1]

15 -1 1 65664 torch.nn.modules.conv.ConvTranspose2d [128, 128, 2, 2]

16 [-1, 4] 1 0 models.common.Concat [1]

17 -1 1 90880 models.common.C3 [256, 128, 1, False]

18 -1 1 147712 models.common.Conv [128, 128, 3, 2]

19 [-1, 14] 1 0 models.common.Concat [1]

20 -1 1 296448 models.common.C3 [256, 256, 1, False]

21 -1 1 590336 models.common.Conv [256, 256, 3, 2]

22 [-1, 10] 1 0 models.common.Concat [1]

23 -1 1 1182720 models.common.C3 [512, 512, 1, False]

24 [17, 20, 23] 1 18879 models.yolo.Detect [2, [[10, 13, 16, 30, 33, 23], [30, 61, 62, 45, 59, 119], [116, 90, 156, 198, 373, 326]], [128, 256, 512]]

/home/user/.local/lib/python3.6/site-packages/torch/functional.py:445: UserWarning: torch.meshgrid: in an upcoming release, it will be required to pass the indexing argument. (Triggered internally at ../aten/src/ATen/native/TensorShape.cpp:2157.)

return \_VF.meshgrid(tensors, \*\*kwargs) # type: ignore[attr-defined]

Model Summary: 282 layers, 7391711 parameters, 7391711 gradients, 17.6 GFLOPS

Transferred 252/366 items from weights/yolov5s.pt

Scaled weight\_decay = 0.0005

Optimizer groups: 64 .bias, 64 conv.weight, 59 other

Scanning '/home/user/YOLO/博客园文档/yolov5-4.0/data/labels' for images and labels... 1 found, 0 missing, 0 empty, 0 corrupted: 100%|██████| 1/1 [00:00<00:00, 155.31it/s]

New cache created: /home/user/YOLO/博客园文档/yolov5-4.0/data/labels.cache

Scanning '/home/user/YOLO/博客园文档/yolov5-4.0/data/labels.cache' for images and labels... 1 found, 0 missing, 0 empty, 0 corrupted: 100%|█████████| 1/1 [00:00<?, ?it/s]

Scanning '/home/user/YOLO/博客园文档/yolov5-4.0/data/labels' for images and labels... 1 found, 0 missing, 0 empty, 0 corrupted: 100%|█████| 1/1 [00:00<00:00, 1250.17it/s]

New cache created: /home/user/YOLO/博客园文档/yolov5-4.0/data/labels.cache

Scanning '/home/user/YOLO/博客园文档/yolov5-4.0/data/labels.cache' for images and labels... 1 found, 0 missing, 0 empty, 0 corrupted: 100%|█████████| 1/1 [00:00<?, ?it/s]

Plotting labels...

Image sizes 640 train, 640 test

Using 0 dataloader workers

Logging results to runs/train/exp13

Starting training for 1 epochs...

Epoch gpu\_mem box obj cls total targets img\_size

0%| | 0/1 [00:00<?, ?it/s]/home/user/.local/lib/python3.6/site-packages/torch/autocast\_mode.py:141: UserWarning: User provided device\_type of 'cuda', but CUDA is not available. Disabling

warnings.warn('User provided device\_type of \'cuda\', but CUDA is not available. Disabling')

0/0 0G 0.07714 0.02705 0.01611 0.1203 4 640: 100%|█████████████████████████████████████████████████████| 1/1 [00:02<00:00, 2.03s/it]

Class Images Targets P R mAP@.5 mAP@.5:.95: 100%|█████████████████████████████████████████| 1/1 [00:00<00:00, 5.73it/s]

all 1 0 0 0 0 0

Exception in thread Thread-5:

Traceback (most recent call last):

File "/usr/lib/python3.6/threading.py", line 916, in \_bootstrap\_inner

self.run()

File "/usr/lib/python3.6/threading.py", line 864, in run

self.\_target(\*self.\_args, \*\*self.\_kwargs)

File "/home/user/YOLO/博客园文档/yolov5-4.0/utils/plots.py", line 124, in plot\_images

colors = color\_list() # list of colors

File "/home/user/YOLO/博客园文档/yolov5-4.0/utils/plots.py", line 34, in color\_list

return [hex2rgb(h) for h in plt.rcParams['axes.prop\_cycle'].by\_key()['color']]

File "/home/user/YOLO/博客园文档/yolov5-4.0/utils/plots.py", line 34, in <listcomp>

return [hex2rgb(h) for h in plt.rcParams['axes.prop\_cycle'].by\_key()['color']]

File "/home/user/YOLO/博客园文档/yolov5-4.0/utils/plots.py", line 32, in hex2rgb

return tuple(int(h[1 + i:1 + i + 2], 16) for i in (0, 2, 4))

File "/home/user/YOLO/博客园文档/yolov5-4.0/utils/plots.py", line 32, in <genexpr>

return tuple(int(h[1 + i:1 + i + 2], 16) for i in (0, 2, 4))

TypeError: int() can't convert non-string with explicit base

Optimizer stripped from runs/train/exp13/weights/last.pt, 15.0MB

Optimizer stripped from runs/train/exp13/weights/best.pt, 15.0MB

1 epochs completed in 0.001 hours.

2.ONNX转化

user@ubuntu:~/YOLO/博客园文档/yolov5-4.0$ python3 models/export.py --weights runs/train/exp12/weights/last.pt

Namespace(batch\_size=1, img\_size=[640, 640], weights='runs/train/exp12/weights/last.pt')

Fusing layers...

/home/user/.local/lib/python3.6/site-packages/torch/functional.py:445: UserWarning: torch.meshgrid: in an upcoming release, it will be required to pass the indexing argument. (Triggered internally at ../aten/src/ATen/native/TensorShape.cpp:2157.)

return \_VF.meshgrid(tensors, \*\*kwargs) # type: ignore[attr-defined]

Model Summary: 223 layers, 7382079 parameters, 0 gradients, 17.4 GFLOPS

Starting TorchScript export with torch 1.10.0+cpu...

/home/user/.local/lib/python3.6/site-packages/torch/jit/\_trace.py:965: TracerWarning: Encountering a list at the output of the tracer might cause the trace to be incorrect, this is only valid if the container structure does not change based on the module's inputs. Consider using a constant container instead (e.g. for `list`, use a `tuple` instead. for `dict`, use a `NamedTuple` instead). If you absolutely need this and know the side effects, pass strict=False to trace() to allow this behavior.

argument\_names,

TorchScript export success, saved as runs/train/exp12/weights/last.torchscript.pt

Starting ONNX export with onnx 1.8.1...

ONNX export success, saved as runs/train/exp12/weights/last.onnx

CoreML export failure: No module named 'coremltools'

Export complete (6.52s). Visualize with https://github.com/lutzroeder/netron.

user@ubuntu:~/YOLO/博客园文档/yolov5-4.0$ python3 -m onnxsim runs/train/exp12/weights/last.onnx runs/train/exp12/weights/yolov5s\_sim.onnx

Installing onnxruntime by `/usr/bin/python3 -m pip install --user onnxruntime`,

please wait for a moment..

Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple

Collecting onnxruntime

Downloading https://pypi.tuna.tsinghua.edu.cn/packages/28/8f/6d175fb62c86d48ae96f170ac9d01118285d7dbb8cded5e35463c96d0ff0/onnxruntime-1.10.0-cp36-cp36m-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (4.9 MB)

|████████████████████████████████| 4.9 MB 5.7 MB/s

Requirement already satisfied: numpy>=1.16.6 in /home/user/.local/lib/python3.6/site-packages (from onnxruntime) (1.19.5)

Requirement already satisfied: protobuf in /home/user/.local/lib/python3.6/site-packages (from onnxruntime) (3.19.6)

Collecting flatbuffers

Downloading https://pypi.tuna.tsinghua.edu.cn/packages/41/f0/7e988a019bc54b2dbd0ad4182ef2d53488bb02e58694cd79d61369e85900/flatbuffers-24.3.25-py2.py3-none-any.whl (26 kB)

Installing collected packages: flatbuffers, onnxruntime

Successfully installed flatbuffers-24.3.25 onnxruntime-1.10.0

Simplifying...

Finish! Here is the difference:

┏━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━┓

┃ ┃ Original Model ┃ Simplified Model ┃

┡━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━┩

│ Add │ 7 │ 7 │

│ Concat │ 13 │ 13 │

│ Conv │ 62 │ 62 │

│ ConvTranspose │ 2 │ 2 │

│ MaxPool │ 3 │ 3 │

│ Relu │ 59 │ 59 │

│ Reshape │ 3 │ 3 │

│ Transpose │ 3 │ 3 │

│ Model Size │ 28.2MiB │ 28.2MiB │

└───────────────┴────────────────┴──────────────────┘

3.NNIE工具安装成功(**nnie mapper**)

user@ubuntu:~/YOLO/博客园文档/软件/依赖/yolov5\_caffe2nnie/data$ nnie\_mapper\_12 classification/alexnet/alexnet\_no\_group\_inst.cfg

Mapper Version 1.2.2.1\_B030 (NNIE\_1.2) 19090610466402

begin net parsing....

end net parsing

begin prev optimizing....

end prev optimizing....

begin net quantalizing(CPU)....

end quantalizing

begin optimizing....

end optimizing

begin NNIE[0] mem allocation....

end NNIE[0] memory allocating

begin NNIE[0] instruction generating....

end NNIE[0] instruction generating

begin parameter compressing....

end parameter compressing

begin compress index generating....

end compress index generating

begin binary code generating....

end binary code generating

begin quant files writing....

end quant files writing

4.onnx转化成caffe

很长很长，反正有这串就是成了，然后去看看有没有生成onnx模型相关的文件

I1103 01:10:32.028074 3935 net.cpp:202] MaxPool\_69 does not need backward computation.

I1103 01:10:32.028079 3935 net.cpp:202] MaxPool\_68 does not need backward computation.

I1103 01:10:32.028084 3935 net.cpp:202] 198\_Relu\_67\_0\_split does not need backward computation.

I1103 01:10:32.028087 3935 net.cpp:202] Relu\_67 does not need backward computation.

I1103 01:10:32.028093 3935 net.cpp:202] Conv\_66 does not need backward computation.

I1103 01:10:32.028097 3935 net.cpp:202] Relu\_65 does not need backward computation.

I1103 01:10:32.028101 3935 net.cpp:202] Conv\_64 does not need backward computation.

I1103 01:10:32.028106 3935 net.cpp:202] 194\_Relu\_63\_0\_split does not need backward computation.

I1103 01:10:32.028111 3935 net.cpp:202] Relu\_63 does not need backward computation.

I1103 01:10:32.028116 3935 net.cpp:202] Conv\_62 does not need backward computation.

I1103 01:10:32.028121 3935 net.cpp:202] Concat\_61 does not need backward computation.

I1103 01:10:32.028126 3935 net.cpp:202] Relu\_60 does not need backward computation.

I1103 01:10:32.028129 3935 net.cpp:202] Conv\_59 does not need backward computation.

I1103 01:10:32.028136 3935 net.cpp:202] Add\_58 does not need backward computation.

I1103 01:10:32.028141 3935 net.cpp:202] Relu\_57 does not need backward computation.

I1103 01:10:32.028144 3935 net.cpp:202] Conv\_56 does not need backward computation.

I1103 01:10:32.028148 3935 net.cpp:202] Relu\_55 does not need backward computation.

I1103 01:10:32.028162 3935 net.cpp:202] Conv\_54 does not need backward computation.

I1103 01:10:32.028167 3935 net.cpp:202] 184\_Add\_53\_0\_split does not need backward computation.

I1103 01:10:32.028172 3935 net.cpp:202] Add\_53 does not need backward computation.

I1103 01:10:32.028177 3935 net.cpp:202] Relu\_52 does not need backward computation.

I1103 01:10:32.028180 3935 net.cpp:202] Conv\_51 does not need backward computation.

I1103 01:10:32.028185 3935 net.cpp:202] Relu\_50 does not need backward computation.

I1103 01:10:32.028195 3935 net.cpp:202] Conv\_49 does not need backward computation.

I1103 01:10:32.028198 3935 net.cpp:202] 179\_Add\_48\_0\_split does not need backward computation.

I1103 01:10:32.028200 3935 net.cpp:202] Add\_48 does not need backward computation.

I1103 01:10:32.028211 3935 net.cpp:202] Relu\_47 does not need backward computation.

I1103 01:10:32.028213 3935 net.cpp:202] Conv\_46 does not need backward computation.

I1103 01:10:32.028223 3935 net.cpp:202] Relu\_45 does not need backward computation.

I1103 01:10:32.028225 3935 net.cpp:202] Conv\_44 does not need backward computation.

I1103 01:10:32.028227 3935 net.cpp:202] 174\_Relu\_43\_0\_split does not need backward computation.

I1103 01:10:32.028237 3935 net.cpp:202] Relu\_43 does not need backward computation.

I1103 01:10:32.028239 3935 net.cpp:202] Conv\_42 does not need backward computation.

I1103 01:10:32.028241 3935 net.cpp:202] 172\_Relu\_41\_0\_split does not need backward computation.

I1103 01:10:32.028252 3935 net.cpp:202] Relu\_41 does not need backward computation.

I1103 01:10:32.028254 3935 net.cpp:202] Conv\_40 does not need backward computation.

I1103 01:10:32.028256 3935 net.cpp:202] 170\_Relu\_39\_0\_split does not need backward computation.

I1103 01:10:32.028265 3935 net.cpp:202] Relu\_39 does not need backward computation.

I1103 01:10:32.028267 3935 net.cpp:202] Conv\_38 does not need backward computation.

I1103 01:10:32.028270 3935 net.cpp:202] Concat\_37 does not need backward computation.

I1103 01:10:32.028280 3935 net.cpp:202] Relu\_36 does not need backward computation.

I1103 01:10:32.028281 3935 net.cpp:202] Conv\_35 does not need backward computation.

I1103 01:10:32.028283 3935 net.cpp:202] Add\_34 does not need backward computation.

I1103 01:10:32.028293 3935 net.cpp:202] Relu\_33 does not need backward computation.

I1103 01:10:32.028297 3935 net.cpp:202] Conv\_32 does not need backward computation.

I1103 01:10:32.028307 3935 net.cpp:202] Relu\_31 does not need backward computation.

I1103 01:10:32.028309 3935 net.cpp:202] Conv\_30 does not need backward computation.

I1103 01:10:32.028311 3935 net.cpp:202] 160\_Add\_29\_0\_split does not need backward computation.

I1103 01:10:32.028321 3935 net.cpp:202] Add\_29 does not need backward computation.

I1103 01:10:32.028324 3935 net.cpp:202] Relu\_28 does not need backward computation.

I1103 01:10:32.028327 3935 net.cpp:202] Conv\_27 does not need backward computation.

I1103 01:10:32.028333 3935 net.cpp:202] Relu\_26 does not need backward computation.

I1103 01:10:32.028338 3935 net.cpp:202] Conv\_25 does not need backward computation.

I1103 01:10:32.028343 3935 net.cpp:202] 155\_Add\_24\_0\_split does not need backward computation.

I1103 01:10:32.028348 3935 net.cpp:202] Add\_24 does not need backward computation.

I1103 01:10:32.028353 3935 net.cpp:202] Relu\_23 does not need backward computation.

I1103 01:10:32.028363 3935 net.cpp:202] Conv\_22 does not need backward computation.

I1103 01:10:32.028366 3935 net.cpp:202] Relu\_21 does not need backward computation.

I1103 01:10:32.028367 3935 net.cpp:202] Conv\_20 does not need backward computation.

I1103 01:10:32.028378 3935 net.cpp:202] 150\_Relu\_19\_0\_split does not need backward computation.

I1103 01:10:32.028379 3935 net.cpp:202] Relu\_19 does not need backward computation.

I1103 01:10:32.028381 3935 net.cpp:202] Conv\_18 does not need backward computation.

I1103 01:10:32.028393 3935 net.cpp:202] 148\_Relu\_17\_0\_split does not need backward computation.

I1103 01:10:32.028394 3935 net.cpp:202] Relu\_17 does not need backward computation.

I1103 01:10:32.028404 3935 net.cpp:202] Conv\_16 does not need backward computation.

I1103 01:10:32.028406 3935 net.cpp:202] Relu\_15 does not need backward computation.

I1103 01:10:32.028409 3935 net.cpp:202] Conv\_14 does not need backward computation.

I1103 01:10:32.028420 3935 net.cpp:202] Concat\_13 does not need backward computation.

I1103 01:10:32.028461 3935 net.cpp:202] Relu\_12 does not need backward computation.

I1103 01:10:32.028470 3935 net.cpp:202] Conv\_11 does not need backward computation.

I1103 01:10:32.028477 3935 net.cpp:202] Add\_10 does not need backward computation.

I1103 01:10:32.028483 3935 net.cpp:202] Relu\_9 does not need backward computation.

I1103 01:10:32.028491 3935 net.cpp:202] Conv\_8 does not need backward computation.

I1103 01:10:32.028496 3935 net.cpp:202] Relu\_7 does not need backward computation.

I1103 01:10:32.028501 3935 net.cpp:202] Conv\_6 does not need backward computation.

I1103 01:10:32.028504 3935 net.cpp:202] 136\_Relu\_5\_0\_split does not need backward computation.

I1103 01:10:32.028509 3935 net.cpp:202] Relu\_5 does not need backward computation.

I1103 01:10:32.028514 3935 net.cpp:202] Conv\_4 does not need backward computation.

I1103 01:10:32.028519 3935 net.cpp:202] 134\_Relu\_3\_0\_split does not need backward computation.

I1103 01:10:32.028523 3935 net.cpp:202] Relu\_3 does not need backward computation.

I1103 01:10:32.028529 3935 net.cpp:202] Conv\_2 does not need backward computation.

I1103 01:10:32.028540 3935 net.cpp:202] Relu\_1 does not need backward computation.

I1103 01:10:32.028542 3935 net.cpp:202] Conv\_0 does not need backward computation.

I1103 01:10:32.028544 3935 net.cpp:202] images does not need backward computation.

I1103 01:10:32.028553 3935 net.cpp:244] This network produces output 301

I1103 01:10:32.028556 3935 net.cpp:244] This network produces output 315

I1103 01:10:32.028558 3935 net.cpp:244] This network produces output output

I1103 01:10:32.028627 3935 net.cpp:257] Network initialization done.

1. 验证caffe模型

user@ubuntu:~/YOLO/博客园文档/yolov5\_caffe-master$ ./build/tools/caffe\_yolov5s

有下面这一长串，最后出现：average time : 548.82 ms 说明caffe模型推理成功

preprocess\_img finished!

[ 0 ] 976 ms.

[ 1 ] 553 ms.

[ 2 ] 535 ms.

[ 3 ] 536 ms.

[ 4 ] 533 ms.

[ 5 ] 534 ms.

[ 6 ] 537 ms.

[ 7 ] 542 ms.

[ 8 ] 537 ms.

[ 9 ] 542 ms.

[ 10 ] 537 ms.

[ 11 ] 547 ms.

[ 12 ] 534 ms.

[ 13 ] 544 ms.

[ 14 ] 543 ms.

[ 15 ] 539 ms.

[ 16 ] 545 ms.

[ 17 ] 535 ms.

[ 18 ] 537 ms.

[ 19 ] 539 ms.

[ 20 ] 537 ms.

[ 21 ] 568 ms.

[ 22 ] 535 ms.

[ 23 ] 539 ms.

[ 24 ] 537 ms.

[ 25 ] 535 ms.

[ 26 ] 536 ms.

[ 27 ] 537 ms.

[ 28 ] 536 ms.

[ 29 ] 539 ms.

[ 30 ] 539 ms.

[ 31 ] 538 ms.

[ 32 ] 537 ms.

[ 33 ] 534 ms.

[ 34 ] 536 ms.

[ 35 ] 539 ms.

[ 36 ] 534 ms.

[ 37 ] 537 ms.

[ 38 ] 534 ms.

[ 39 ] 542 ms.

[ 40 ] 545 ms.

[ 41 ] 555 ms.

[ 42 ] 539 ms.

[ 43 ] 541 ms.

[ 44 ] 542 ms.

[ 45 ] 537 ms.

[ 46 ] 545 ms.

[ 47 ] 546 ms.

[ 48 ] 545 ms.

[ 49 ] 552 ms.

average time : 548.82 ms