

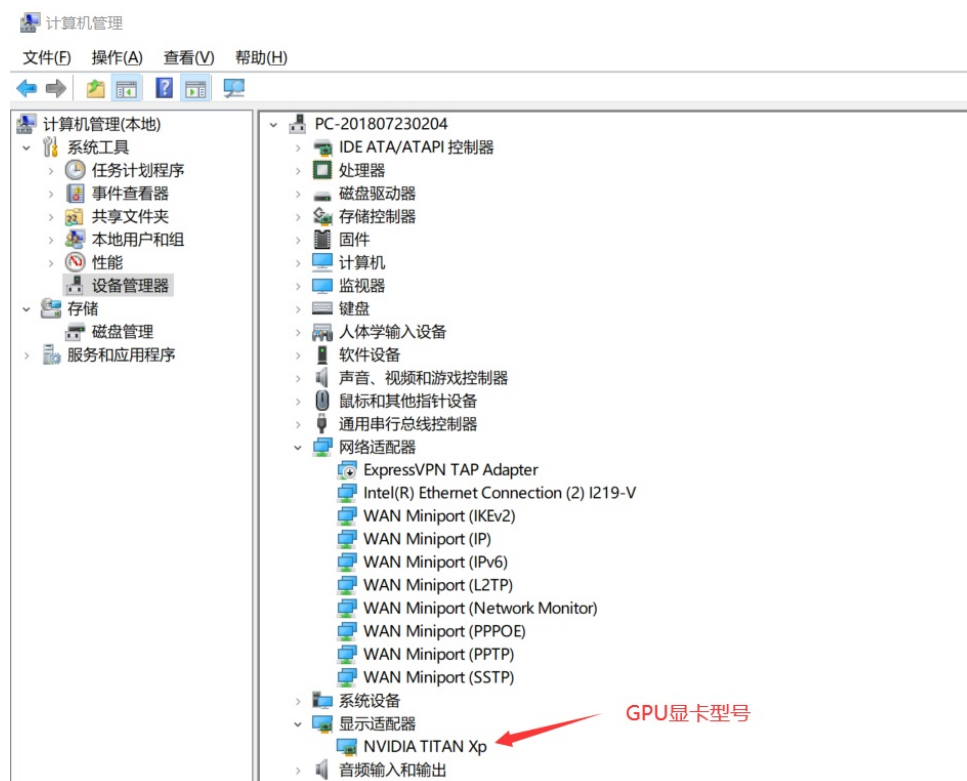
# YOLOv5训练PASCAL VOC数据集

课程演示环境：Windows10

## 1 安装Nvidia显卡驱动、CUDA和cuDNN

### 1) 下载和安装nvidia显卡驱动

首先要在设备管理器中查看你的显卡型号，比如在这里可以看到我的显卡型号为Titan XP。



NVIDIA 驱动下载: <https://www.nvidia.cn/Download/index.aspx?lang=cn>

下载对应你的英伟达显卡驱动。

### 驱动程序下载

NVIDIA > 驱动程序下载

### NVIDIA 驱动程序下载

选项 1: 手动查找适用于我的 NVIDIA 产品的驱动程序。

产品类型:	TITAN	▼
产品系列:	NVIDIA TITAN Series	▼
产品家族:	NVIDIA TITAN Xp	▼
操作系统:	Windows 10 64-bit	▼
下载类型:	Studio 驱动程序 (SD)	▼ ?
语言:	Chinese (Simplified)	▼

下载之后就是简单的下一步直到完成。

完成之后，在cmd中输入执行：

nvidia-smi

如果有错误:

'nvidia-smi' 不是内部或外部命令, 也不是可运行的程序 或批处理文件。

把C:\Program Files\NVIDIA Corporation\NVSMI添加到环境变量的path中。再重新打开cmd窗口。

如果输出下图所示的显卡信息, 说明你的驱动安装成功。

```

c:\ Windows 命令处理程序
Microsoft Windows [版本 10.0.17134.472]
(c) 2018 Microsoft Corporation. 保留所有权利。

C:\Windows\System32>nvidia-smi
Thu Apr 30 17:22:50 2020

+-----+
| NVIDIA-SMI 441.22             Driver Version: 441.22          CUDA Version: 10.2     |
+-----+-----+
| GPU   Name           TCC/WDDM  Bus-Id        Disp.A   Volatile Uncorr. ECC  |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage  GPU-Util  Compute M. |
+-----+-----+
|  0    TITAN Xp         WDDM     00000000:65:00:0    On           1%           N/A  |
| 23%   26C    P8      11W / 250W  | 727MiB / 12288MiB             Default  |
+-----+-----+

+-----+
| Processes:                      GPU Memory |
|   GPU       PID    Type    Process name      Usage  |
+-----+-----+
|    0         1160    C+G    Insufficient Permissions  N/A    |
|    0         7316    C+G    C:\Windows\explorer.exe  N/A    |
|    0         7412    C+G    ...t_cw5nlh2txyewy\ShellExperienceHost.exe N/A    |
|    0         7624    C+G    C:\Program Files\Typora\Typora.exe  N/A    |
|    0         8220    C+G    ...dows.Cortana_cw5nlh2txyewy\SearchUI.exe N/A    |
|    0         9064    C+G    ...hell.Experiences.TextInput.InputApp.exe N/A    |
|    0        11856    C+G    ...ta\Roaming\360se6\Application\360se.exe N/A    |
+-----+

```

## 2) 下载CUDA

本人安装的是CUDA的10.2版本

cuda下载链接: [https://developer.nvidia.com/cuda-downloads?](https://developer.nvidia.com/cuda-downloads?target_os=Windows&target_arch=x86_64&target_version=10&target_type=exelocal)

[target\\_os=Windows&target\\_arch=x86\\_64&target\\_version=10&target\\_type=exelocal](https://developer.nvidia.com/cuda-downloads?target_os=Windows&target_arch=x86_64&target_version=10&target_type=exelocal)

**NVIDIA. HIGH PERFORMANCE COMPUTING**

[DOWNLOADS](#)
[TRAINING](#)
[ECOSYSTEM](#)
[FORUMS](#)

### CUDA Toolkit 10.2 Download

Join us for GTC Digital on Thursday, March 26th, where we will host a full-day, instructor-led, online workshop covering the "Fundamentals of CUDA Programming". [Register at our significantly discounted rate \[\\$79\].](#)

[Home](#) > [High Performance Computing](#) > [CUDA Toolkit](#) > [CUDA Toolkit 10.2 Download](#)

### Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System

Windows

Linux

Mac OSX

Architecture

x86\_64

Version

10

8.1

7

Server 2019

Server 2016

Server 2012 R2

Installer Type

exe (network)

exe (local)

下载后得到文件: cuda\_10.2.89\_441.22\_win10.exe

### 3) 下载cuDNN

cuda下载地址: <https://developer.nvidia.com/cudnn>

需要有账号

## cuDNN Download

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

☒ I Agree To the Terms of the [cuDNN Software License Agreement](#)

Note: Please refer to the [Installation Guide](#) for release prerequisites, including supported GPU architectures and compute capabilities, before downloading.

For more information, refer to the cuDNN Developer Guide, Installation Guide and Release Notes on the [Deep Learning SDK Documentation](#) web page.

Download cuDNN v7.6.5 (November 18th, 2019), for CUDA 10.2

### Library for Windows, Mac, Linux, Ubuntu and RedHat/Centos(x86\_64architecture)

cuDNN Library for Windows 7

cuDNN Library for Windows 10

cuDNN Library for Linux

cuDNN Runtime Library for Ubuntu18.04 (Deb)

cuDNN Developer Library for Ubuntu18.04 (Deb)

cuDNN Code Samples and User Guide for Ubuntu18.04 (Deb)

下载后得到文件: cudnn-10.2-windows10-x64-v7.6.5.32.zip

### 4) 安装cuda

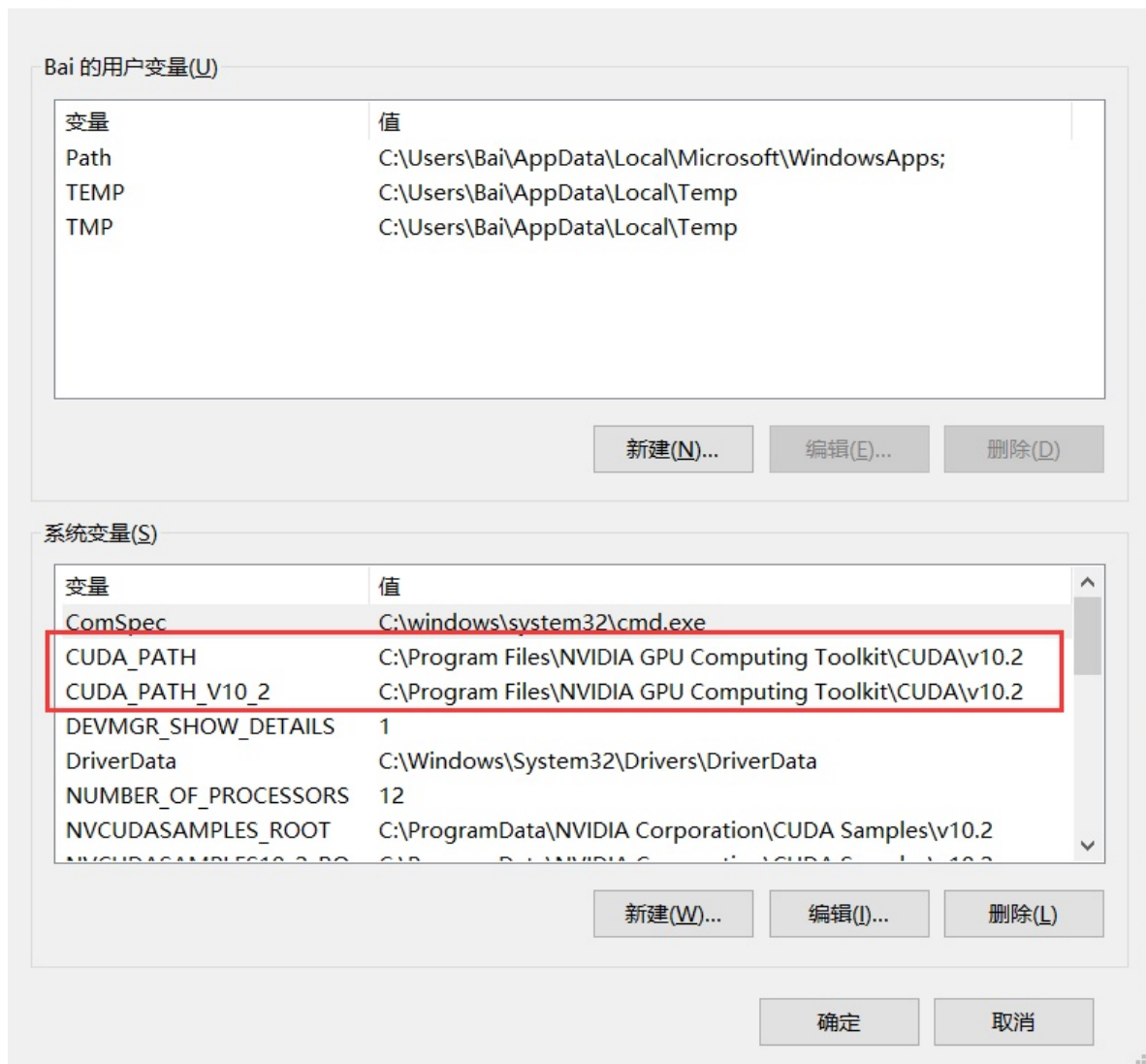
#### (1) 将cuda运行安装, 建议默认路径





安装时可以勾选Visual Studio Integration

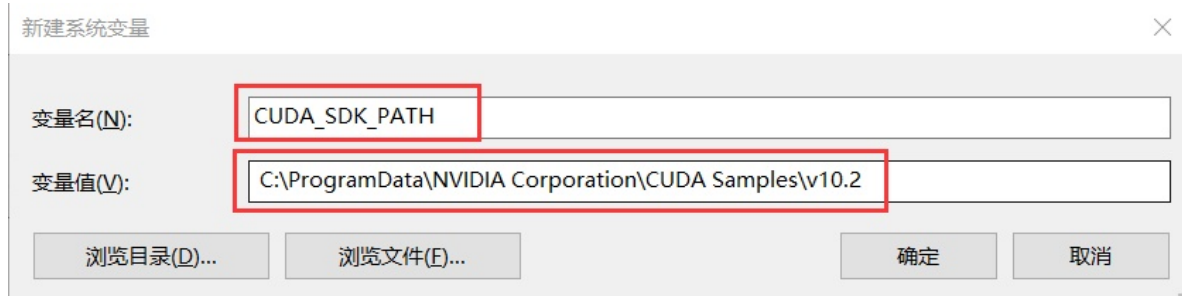
## (2) 安装完成后设置环境变量



计算机上点右键，打开属性->高级系统设置->环境变量，可以看到系统中多了CUDA\_PATH和CUDA\_PATH\_V10\_2两个环境变量。

接下来，还要在系统中添加以下几个环境变量：这是默认安装位置的路径：C:\ProgramData\NVIDIA Corporation\CUDA Samples\v10.2

CUDA\_SDK\_PATH = C:\ProgramData\NVIDIA Corporation\CUDA Samples\v10.2  
CUDA\_LIB\_PATH = %CUDA\_PATH%\lib\x64  
CUDA\_BIN\_PATH = %CUDA\_PATH%\bin  
CUDA\_SDK\_BIN\_PATH = %CUDA\_SDK\_PATH%\bin\win64  
CUDA\_SDK\_LIB\_PATH = %CUDA\_SDK\_PATH%\common\lib\x64



在系统变量 Path 的末尾添加：

%CUDA\_LIB\_PATH%;%CUDA\_BIN\_PATH%;%CUDA\_SDK\_LIB\_PATH%;%CUDA\_SDK\_BIN\_PATH%;

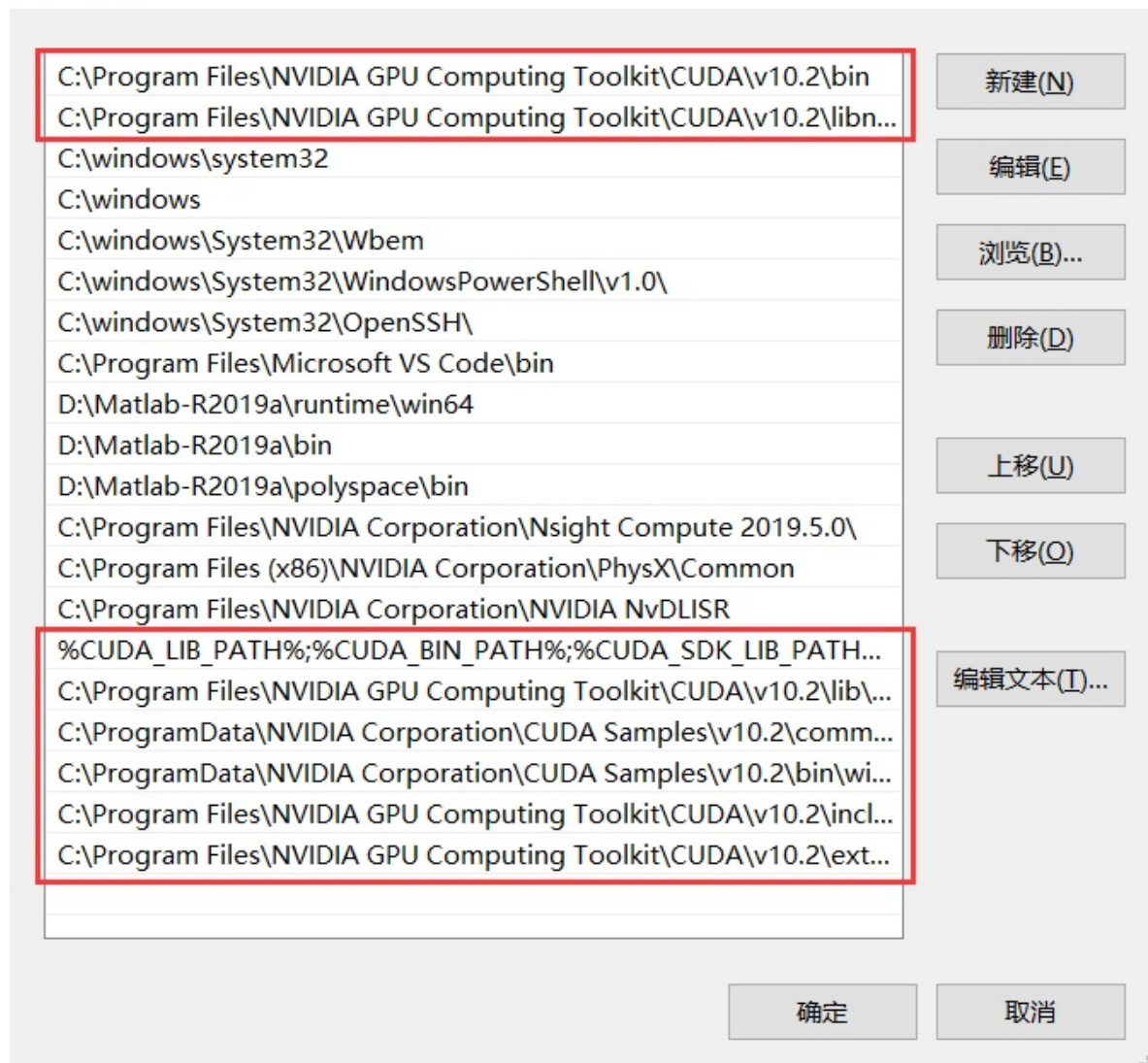
再添加如下5条（默认安装路径）：



C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.2\lib\x64 C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.2\include C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.2\extras\CUPTI\lib64 C:\ProgramData\NVIDIA Corporation\CUDA Samples\v10.2\bin\win64 C:\ProgramData\NVIDIA Corporation\CUDA Samples\v10.2\common\lib\x64

编辑环境变量

×



## 5) 安装cuDNN

### 复制cudnn文件

对于cudnn直接将其解开压缩包，然后需要将bin,include,lib中的文件复制粘贴到cuda的文件夹下

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.2

## 6) CUDA安装测试

最后测试cuda是否配置成功：

打开CMD执行：

```
nvcc -V
```

即可看到cuda的信息

```
C:\Users\Bai>nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2019 NVIDIA Corporation
Built on Wed_Oct_23_19:32:27_Pacific_Daylight_Time_2019
Cuda compilation tools, release 10.2, V10.2.89
```

## 2 安装Anaconda和PyTorch

### 1) 安装Anaconda

Anaconda 是一个用于科学计算的 Python 发行版，支持 Linux, Mac, Windows, 包含了众多流行的科学计算、数据分析的 Python 包。

#### 1) 下载安装包

Anaconda下载Windows版: <https://www.anaconda.com/products/individual>

#### 2) 然后安装anaconda

#### 3) 添加Anaconda国内镜像配置

清华TUNA提供了 Anaconda 仓库的镜像，运行以下命令：

```
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/
```

```
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/
```

```
conda config --set show_channel_urls yes
```

### 2) 安装pytorch

注意：需要安装pytorch 1.6以上的版本 创建虚拟环境，环境名字可自己确定，这里使用yolov5作为环境名：

```
conda create -n yolov5 python=3.8
```

安装成功后激活yolov5环境：

```
conda activate yolov5
```

在所创建的环境下安装pytorch的1.6以上版本, 执行命令：

```
conda install pytorch torchvision cudatoolkit=10.2 -c pytorch
```

注意：10.2处应为自己电脑上的cuda版本号

离线安装：

下载网址: <https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/pytorch/win-64/>

版本: pytorch-1.6.0-py3.8\_cuda102\_cudnn7\_0.tar.bz2

```
conda install --offline pytorch-1.6.0-py3.8_cuda102_cudnn7_0.tar.bz2
```

## 3 yolov5项目克隆和安装

### 1) 克隆yolov5项目

安装Git软件 (<https://git-scm.com/downloads>) , 克隆项目到本地 (如d:)

```
git clone https://github.com/ultralytics/yolov5.git
```

或手动下载yolov5源码v3.1:

<https://github.com/ultralytics/yolov5/releases/tag/v3.1>

### 2) 安装所需库

使用清华镜像源:

在yolov5路径下执行:

```
pip install -r requirements.txt -i https://pypi.tuna.tsinghua.edu.cn/simple
```

注意: simple 不能少, 是 https 而不是 http

### 3) 下载预训练权重文件

下载yolov5s.pt, yolov5m.pt, yolov5l.pt, yolov5x.pt权重文件, 并放置在weights文件夹下

百度网盘下载链接:

链接: <https://pan.baidu.com/s/1RztRLwk3Dye9gkmXxG40JA> 提取码: e2q0

### 4) 安装测试

测试图片:

在yolov5路径下执行

```
python detect.py --source ./inference/images/ --weights weights/yolov5s.pt --conf 0.4
```

## 4 准备数据集

### 1) 下载项目文件:

从百度网盘下载到yolov5目录下并解压

- VOCtrainval\_06-Nov-2007.tar
- VOCtrainval\_11-May-2012.tar
- VOCtest\_06-Nov-2007.tar
- get\_voc\_windows.py

### 2) 生成训练集和验证集文件



执行python脚本：

```
python get_voc_windows.py
```

在VOCdevkit / VOC2007和VOCdevkit / VOC2012目录下可以看到生成了文件夹labels；在VOC目录下生成了images和labels文件夹；

在yolov5目录下生成了文件2007\_train.txt, 2007\_val.txt, 2007\_test.txt, 2012\_train.txt, 2012\_val.txt, train.txt, train.all.txt。

- labels下的文件是JPEGImages文件夹下每一个图像的yolo格式的标注文件，这是由annotations的xml标注文件转换来的
- train.txt和2007\_test.txt分别给出了训练集图片和验证集图片的列表，含有每个图片的路径和文件名。
- VOC/images文件夹下有train和val文件夹，分别放置训练集和验证集图片；VOC/labels文件夹有train和val文件夹，分别放置训练集和验证集标签（yolo格式）。

## 5 修改配置文件

### 1) 新建文件data/voc-new.yaml

可以复制data/voc.yaml再根据自己情况的修改；可以重新命名如：data/voc-new.yaml

然后修改配置参数

```
# download command/URL (optional)
#download: bash data/scripts/get_voc.sh

# train and val data as 1) directory: path/images/, 2) file: path/images.txt, or
# 3) list: [path1/images/, path2/images/]
train: VOC/images/train/ # 16551 images
val: VOC/images/val/ # 4952 images
```

### 2) 新建文件models/yolov5s-voc.yaml

可以复制models/yolov5s.yaml再根据自己情况的修改；可以重新命名如：models/yolov5s-voc.yaml

然后修改配置参数

```
# parameters
nc: 20 # number of classes
```

## 6 训练自己的数据集

### 1) 训练命令

在yolov5路径下执行

```
python train.py --data data/voc-new.yaml --cfg models/yolov5s-voc.yaml --weights
weights/yolov5s.pt --batch-size 16 --epochs 200
```

### 2) 训练过程可视化：

在yolov5路径下执行

```
tensorboard --logdir=./runs
```

## 7 测试训练出的网络模型

### 1) 测试图片

在yolov5路径下执行

```
python detect.py --source ./VOC/images/val/000001.jpg --weights  
runs/exp0/weights/best.pt --conf 0.4
```

### 2) 性能统计

在yolov5路径下执行

```
python test.py --data data/voc-new.yaml --weights runs/exp0/weights/best.pt --  
batch-size 16
```

## 8 导出ONNX文件

```
pip install onnx>=1.7.0 -i https://pypi.tuna.tsinghua.edu.cn/simple # for ONNX  
export  
pip install coremltools==4.0 -i https://pypi.tuna.tsinghua.edu.cn/simple # for  
CoreML export
```

```
python models/export.py --weights weights/yolov5s.pt --img 640 --batch 1 #  
export at 640x640 with batch size 1
```

# 声明

本课程的数据集、程序文件以及课件的演示文稿、视频由讲师白勇拥有知识产权的权利。只限于学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造。

