

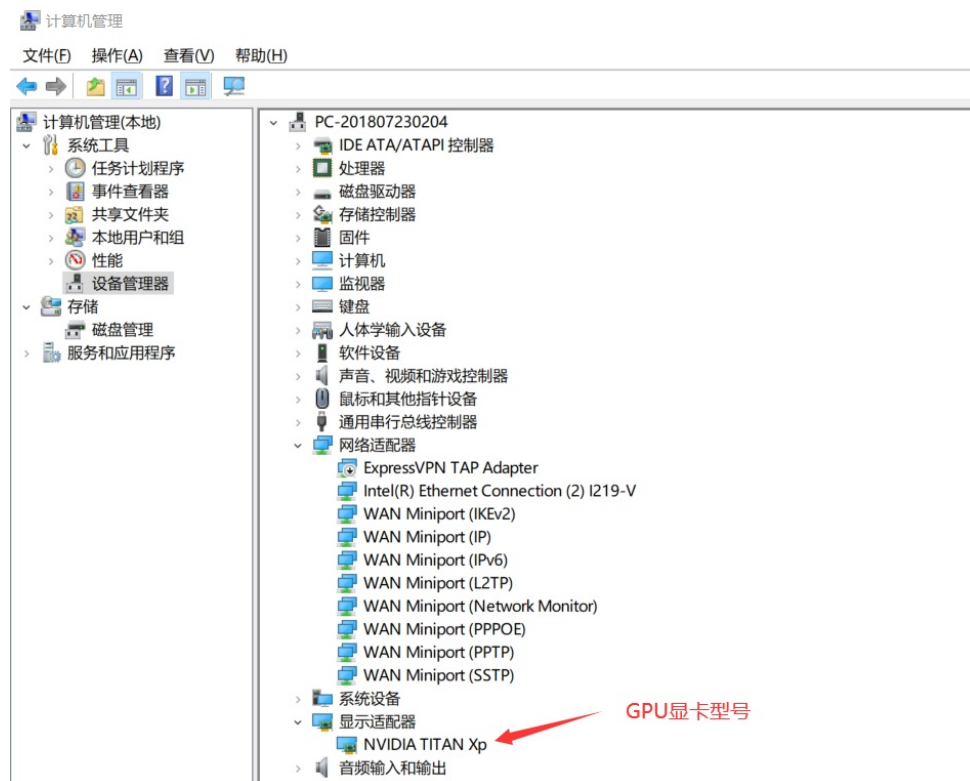
# YOLOv4项目实战-单目标检测：足球

课程演示环境：Windows10

## 1 软件安装

### 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

CUDA目前最新的是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)

## 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

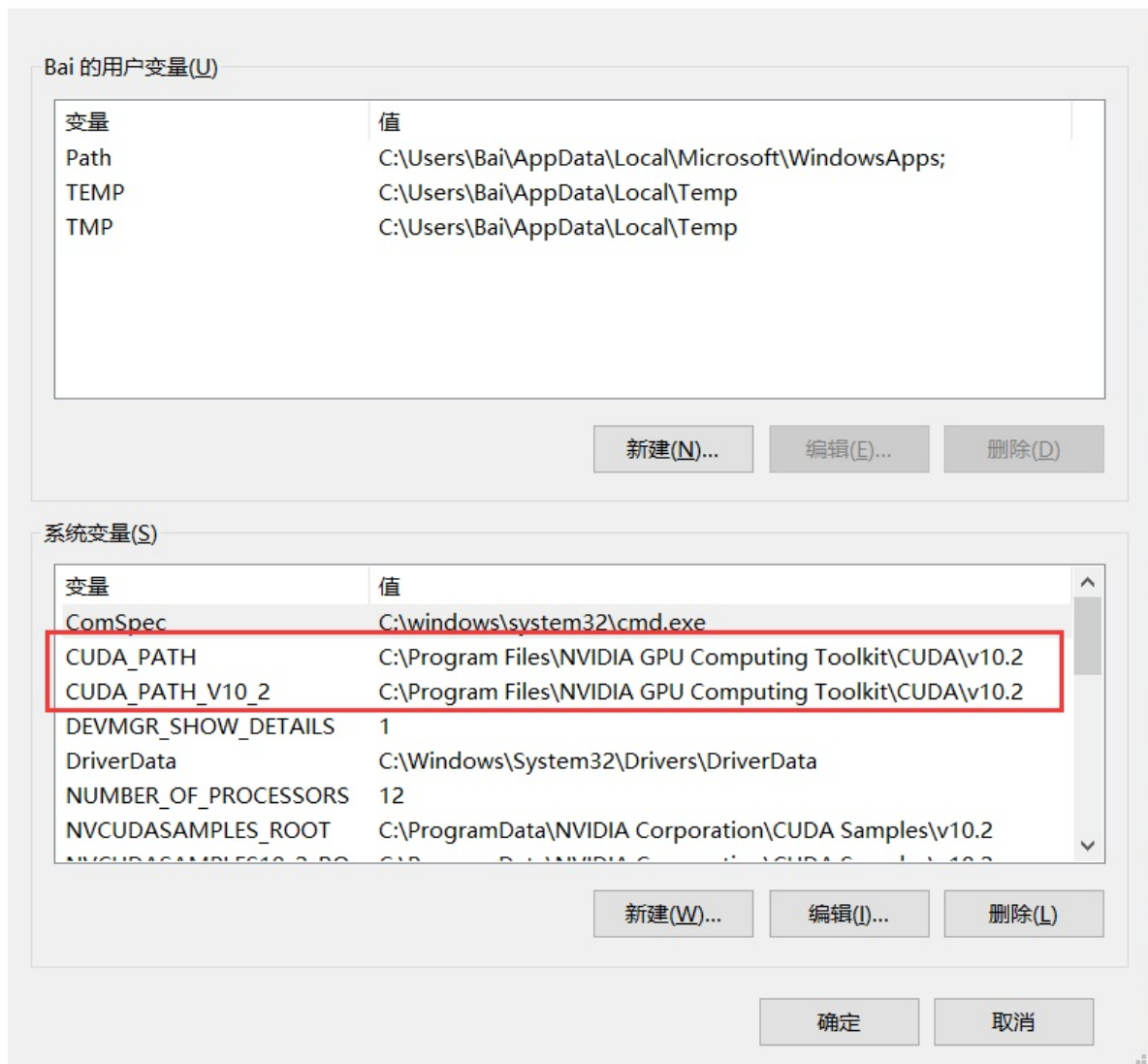
## 4) 安装cuda

(1) 将cuda运行安装, 建议默认路径



安装时可以勾选Visual Studio Integration

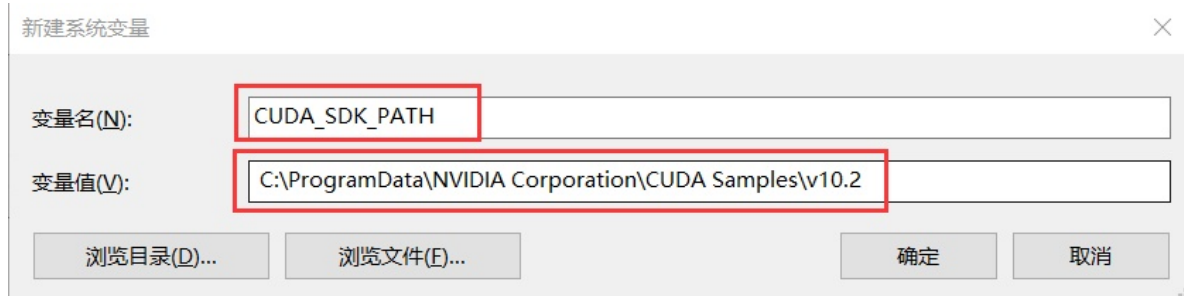
## (2) 安装完成后设置环境变量



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%;

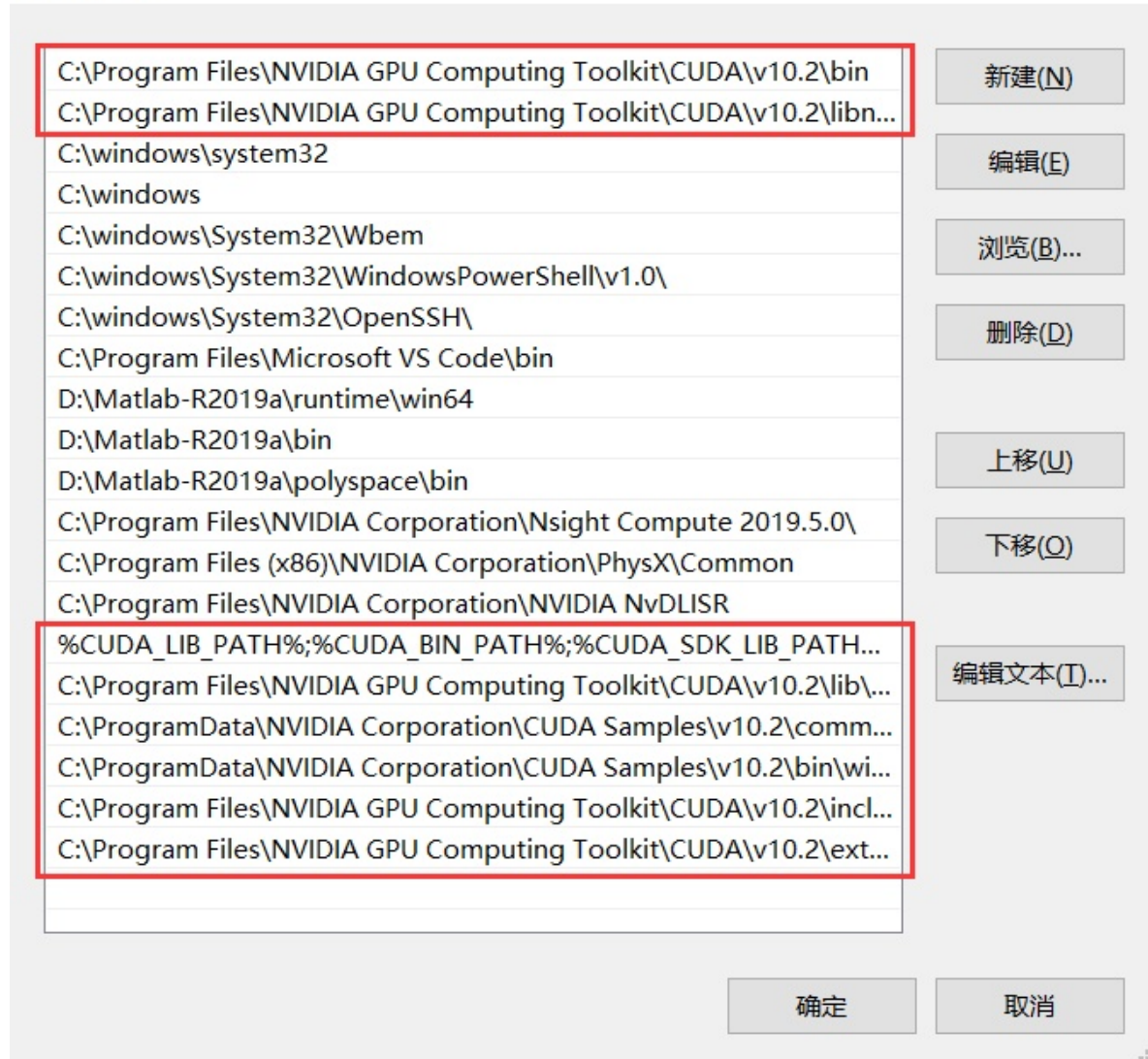
再添加如下5条（默认安装路径）：



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
```

## 7) 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
```

## 8) 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, 这里本人使用pytorch1.6作为环境名:

```
conda create -n pytorch1.6 python=3.8
```

安装成功后激活pytorch1.6环境:

```
conda activate pytorch1.6
```

在所创建的pytorch环境下安装pytorch的1.6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
```

## 2 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/ultraalytics/yolov5.git
```

### 2) 安装所需库

使用清华镜像源:

在yolov5路径下执行:

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

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/1GYlG8Gj0rn90t\\_kUHytGXA](https://pan.baidu.com/s/1GYlG8Gj0rn90t_kUHytGXA) 提取码: udnm

### 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
```

测试视频:

在yolov5路径下执行

```
python detect.py --source driving.mp4 --weights weights/yolov5s.pt
```

## 3. 标注自己的数据集

### 1) 安装图像标注工具labellmg

网址: <https://github.com/tzutalin/labellmg>

下载后得到文件labellmg-master.zip

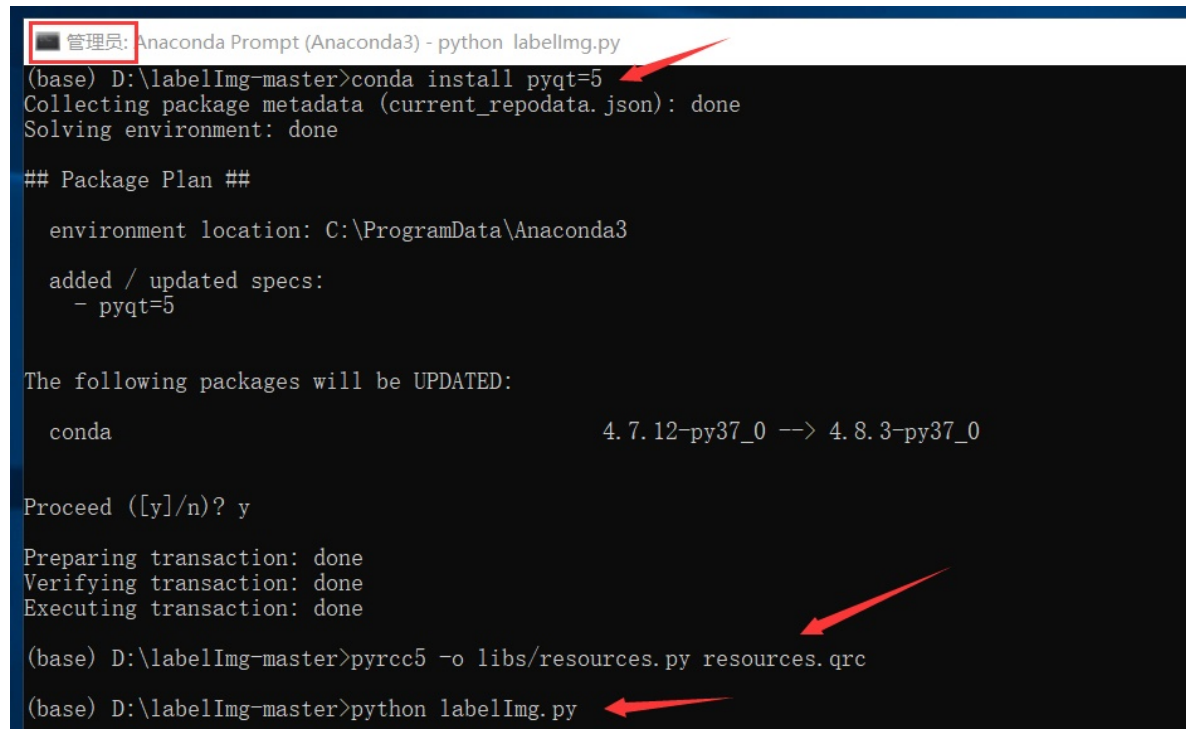
解压: D:\labellmg-master



## 建议使用Anaconda安装

以管理员身份运行Anaconda Prompt并到labelImg-master目录下执行命令

```
conda install pyqt=5
pyrcc5 -o libs/resources.py resources.qrc
python labelImg.py
```



```
管理员: Anaconda Prompt (Anaconda3) - python labelImg.py
(base) D:\labelImg-master>conda install pyqt=5
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\ProgramData\Anaconda3

  added / updated specs:
    - pyqt=5

The following packages will be UPDATED:

  conda                                4.7.12-py37_0 --> 4.8.3-py37_0

Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(base) D:\labelImg-master>pyrcc5 -o libs/resources.py resources.qrc

(base) D:\labelImg-master>python labelImg.py
```

## 2) 添加自定义类别

修改文件labelImg/data/predefined\_classes.txt

```
ball
messi
trophy
```

## 3) 使用labelImg进行图像标注

用labelImg标注生成PASCAL VOC格式的xml标记文件。例如：



width = 1000

height = 654

PASCAL VOC标记文件如下：

```

<?xml version="1.0" encoding="ISO-8859-1"?>
- <annotation>
  <folder>bai</folder>
  <filename>trophy.jpg</filename>
  <path>/home/bai/trophy.jpg</path>
  - <source>
    <database>Unknown</database>
  </source>
  - <size>
    <width>1000</width>
    <height>654</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  - <object>
    <name>trophy</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    - <bndbox>
      <xmin>187</xmin>
      <ymin>21</ymin>
      <xmax>403</xmax>
      <ymax>627</ymax>
    </bndbox>
  </object>
</annotation>

```

也可以直接生成YOLO格式的txt标记文件如下：

**class\_id x y w h**

```
2 0.295000 0.495413 0.216000 0.926606
```

$x = x\_center / width = 295 / 1000 = 0.2950$

$y = y\_center / height = 324 / 654 = 0.4954$

$w = (xmax - xmin) / width = 216 / 1000 = 0.2160$

$h = (ymax - ymin) / height = 606 / 654 = 0.9266$

**class\_id:** 类别的id编号

**x:** 目标的中心点x坐标（横向）/图片总宽度

**y:** 目标的中心的y坐标（纵向）/图片总高度

**w:** 目标框的宽带/图片总宽度

**h:** 目标框的高度/图片总高度

可以用python代码实现两种标记格式的转换：

```
def convert(size, box):  
    dw = 1./size[0]  
    dh = 1./size[1]  
    x = (box[0] + box[1])/2.0  
    y = (box[2] + box[3])/2.0  
    w = box[1] - box[0]  
    h = box[3] - box[2]  
    x = x*dw  
    w = w*dw  
    y = y*dh  
    h = h*dh  
    return (x,y,w,h)
```

box[0]: xmin

box[1]: xmax

box[2]: ymin

box[3]: ymax

### 3 准备自己的数据集

1) 下载项目文件:

从百度网盘下载到yolov5目录下并解压

**百度网盘下载链接:**

链接: [https://pan.baidu.com/s/1GYIG8Gj0rn90t\\_kUHytGXA](https://pan.baidu.com/s/1GYIG8Gj0rn90t_kUHytGXA) 提取码: udnm

- VOCdevkit\_ball.tar.gz
- testfiles.tar.gz
- prepare\_data.py

2) 解压建立或自行建立数据集

使用PASCAL VOC数据集的目录结构:

文件夹层次为 yolov5 / VOCdevkit / VOC2007

VOC2007下面建立两个文件夹: Annotations和JPEGImages

JPEGImages放所有的训练和测试图片; Annotations放所有的xml标记文件

3) 生成训练集和验证集文件

执行python脚本:

```
python prepare_data.py
```

在VOCdevkit / VOC2007目录下可以看到生成了文件夹YOLOLabels; 在VOCdevkit目录下生成了images和labels文件夹; 在yolov5下生成了两个文件yolov5\_train.txt和yolov5\_val.txt。

- YOLOLabels下的文件是images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
- images文件夹下有train和val文件夹, 分别放置训练集和测试集图片; labels文件夹有train和val文件夹, 分别放置训练集和测试集标签 (yolo格式)

- yolov5\_train.txt和yolov5\_val.txt分别给出了训练图片文件和测试图片文件的列表，含有每张图片的路径和文件名。

## 4 修改配置文件

### 1) 新建文件data/voc-ball.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-ball.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: ../yolov5/VOCdevkit/images/train/
val: ../yolov5/VOCdevkit/images/val/

# number of classes
nc: 1

# class names
names: ['ball']
```

### 2) 新建文件models/yolov5s-ball.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-ball.yaml

然后修改配置参数

```
# parameters
nc: 1 # number of classes
```

## 5 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-ball.yaml --cfg models/yolov5s-ball.yaml --
weights weights/yolov5s.pt --batch-size 16 --epochs 50
```

### 2) 训练过程可视化：

在yolov5路径下执行

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

## 6 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 ./testfiles/img1.jpg --weights  
runs/exp0/weights/best.pt --conf 0.4
```

## 2) 测试视频

在yolov5路径下执行

```
python detect.py --source ./testfiles/messi.mp4 --weights  
runs/exp0/weights/best.pt --conf 0.4
```

## 3) 性能统计

在yolov5路径下执行

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