**YOLO V3（GPU）环境的搭建步骤笔记**

参考：

<https://blog.csdn.net/zong596568821xp/article/details/86467456>

<https://blog.csdn.net/weixin_43717579/article/details/85108925>

**创建环境：**

Conda create -n yolov3GPU python=3.6

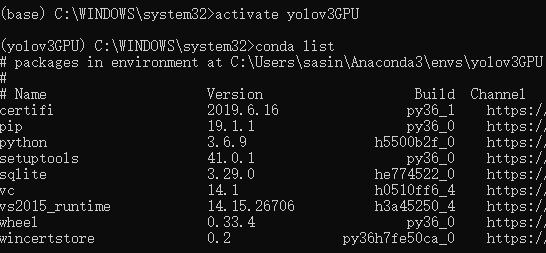
**更换conda源：**

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/

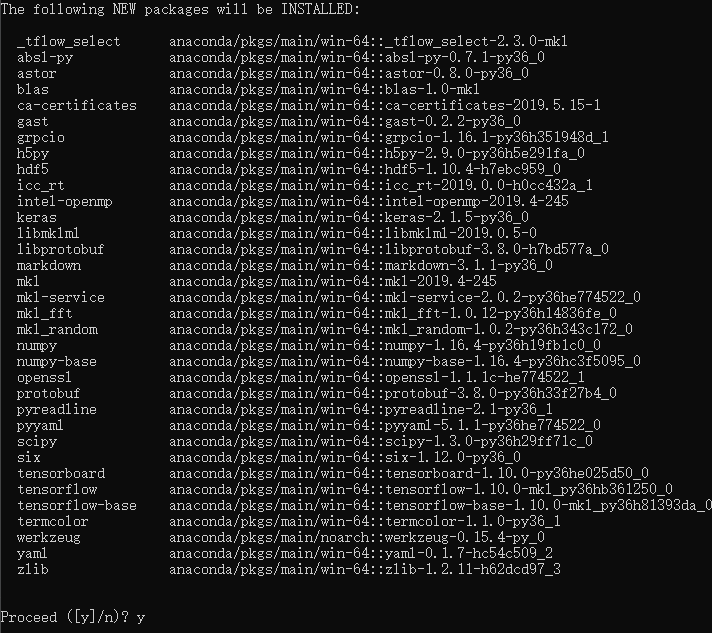
conda config --set show\_channel\_urls yes

**激活环境：**



**安装keras\tensorflow**





**安装tensorflow-GPU:**

pip install -i https://pypi.tuna.tsinghua.edu.cn/simple tensorflow-gpu==1.6.0

-i 指定国内源，要不下载速度太慢。

**安装依赖：**

pip install -i https://pypi.tuna.tsinghua.edu.cn/simple pillow

pip install -i https://pypi.tuna.tsinghua.edu.cn/simple matplotlib

pip install -i https://pypi.tuna.tsinghua.edu.cn/simple ffmpeg

pip install -i https://pypi.tuna.tsinghua.edu.cn/simple opencv-contrib-python

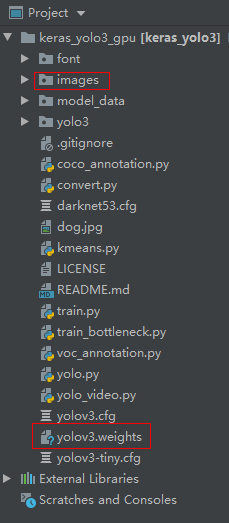
**下载yolov3及权重文件:**

<https://github.com/qqwweee/keras-yolo3>

<https://pjreddie.com/media/files/yolov3.weights>

**创建工程项目：**

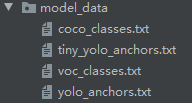
最终目录结构：



Yolov3.weights权重是下载后放入该目录的，images是新建目录，用于存放测试用的图片和视频文件。

### 将将darknet下的配置文件转换成keras适用的h5文件

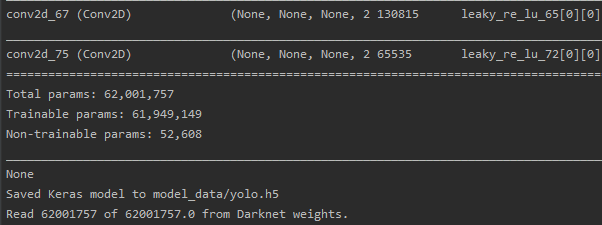
转换前model\_data目录：



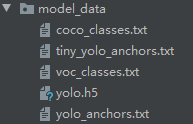
转换命令行：

pytharm环境的最下方，选择“terminal”,在其命令行下，输入如下：

**python convert.py yolov3.cfg yolov3.weights model\_data/yolo.h5**



转换后的mode\_data目录：



**测试图像：**

命令行： python yolo-video.py --image

**报错咯，如下所示：**

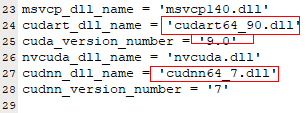
ImportError: Could not find 'cudart64\_90.dll'. TensorFlow requires that this DLL be installed in a directory that is named in your %PATH% environment variable. Download and install CUDA 9.0 from this URL: https://developer.nvidia.com/cuda-toolkit

嗯，么有安装CUDA，查看当前环境需要的CUDA版本。

本环境建立的路径如下：



在上述路径下找到文件：build\_info.py



根据该文件里的版本要求，安装CUDA 和CUDNN即可。

查了一下，tensorflow-gpu 版本高点，会自动安装CUDA 和CUDNN

(**测试后，好像没有自动安装哦**)。

**升级一下tensorflow-gpu:**

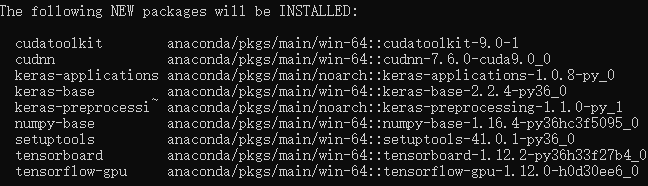


**pip install -i https://pypi.tuna.tsinghua.edu.cn/simple tensorflow-gpu==1.14.0**

**更换版本为1.12.0**

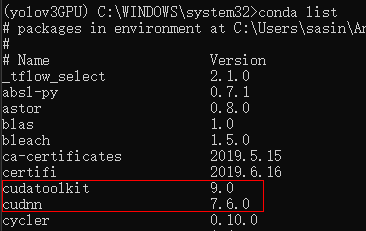
**安装cuda 和 cudnn(conda命令行)**

**Conda install tensorflow-gpu==1.12.0**, 时，**自动安装了 cuda 和 cudnn**，如下所示：



**说明：conda 安装时，比 pip 安装的依赖库要多了。**

**Conda list 查看安装的库：**



**继续测试图像：**

**Python yolo\_video.py --image**

**依然报错：**

tensorflow.python.framework.errors\_impl.InternalError: cudaGetDevice() failed. Status: CUDA driver version is insufficient for CUDA runtime version

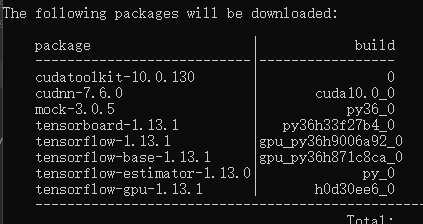
应该是版本的问题。

**升级一下看看：**

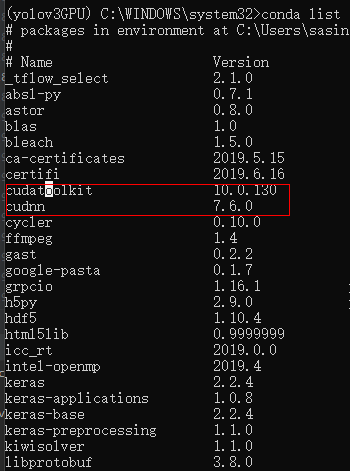
Conda update -all



看提示，cuda and cudnn都升级了版本要。



升级后的cuda 和 cudnn版本如下：

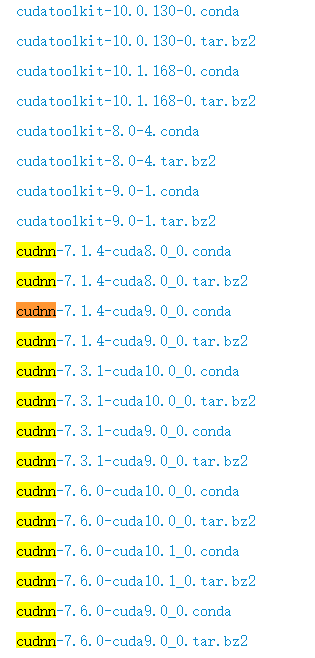


测试图片时，还是报错：

tensorflow.python.framework.errors\_impl.InternalError: cudaGetDevice() failed. Status: CUDA driver version is insufficient for CUDA runtime version

**查看镜像上 的版本：**

<https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/win-64/>



**通过对比，cudnn7.6以上版本，不匹配CUDA9.0哦，cudnn降低版本试试。**

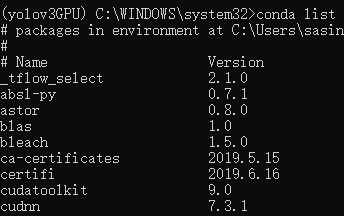
**安装 cudnn7.3.1**

Conda install cudnn==7.3.1





再次查看当前环境cudnn cuda 的版本：



还是同样的错...

**最终解决方案：**

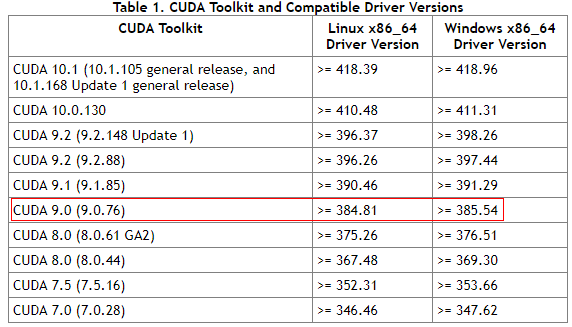
可能是显卡驱动版本问题？查看本机显卡的驱动的版本：



**查看CUDA与驱动的对应关系：**

参考网址：

<https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html>



当前CUDAtoolkit的版本是9.0，对应表中驱动版本要大于等于385.54，，而驱动的版本是382.05，显然不匹配呀。

终极解决方案：

1 要么升级驱动

驱动地址：<https://www.nvidia.cn/Download/index.aspx?lang=cn>

2 要么安装低版本的CUDA，比如安装CUDA8 也是可以的。

**向前看吧，升级驱动，方便以后升级版本。**

用上面描述的驱动地址的连接，根据自己电脑选择后，再下载相应的驱动。安装驱动的过程一直确定下来就可以了。

安装驱动后，再次查看当前的机器的显卡驱动：



对比上面的表，选择余地就大了。

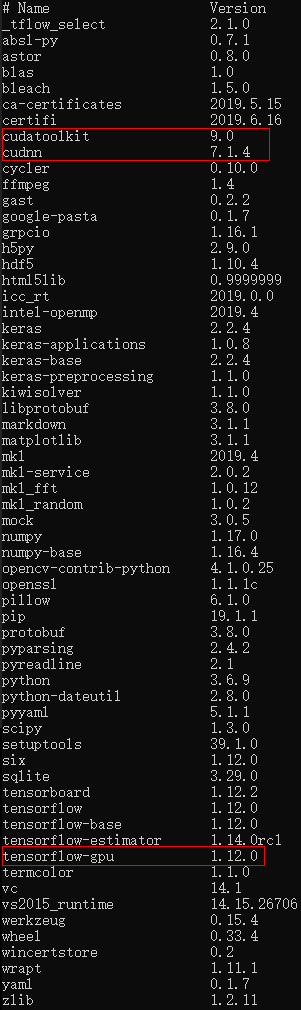
**最后选择的版本：**

**Conda install cudnn==7.3.1**

**Conda install cuda==9.0**

**Conda install tensorflow-gpu==1.12.0**

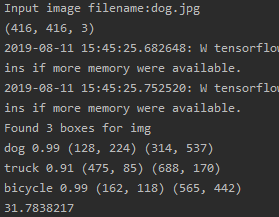
**再次查看当前环境的cuda and cudnn及tensorflow-gpu的版本：**



OK，继续。

**图片测试：**

**Python yolo\_video.py --image**

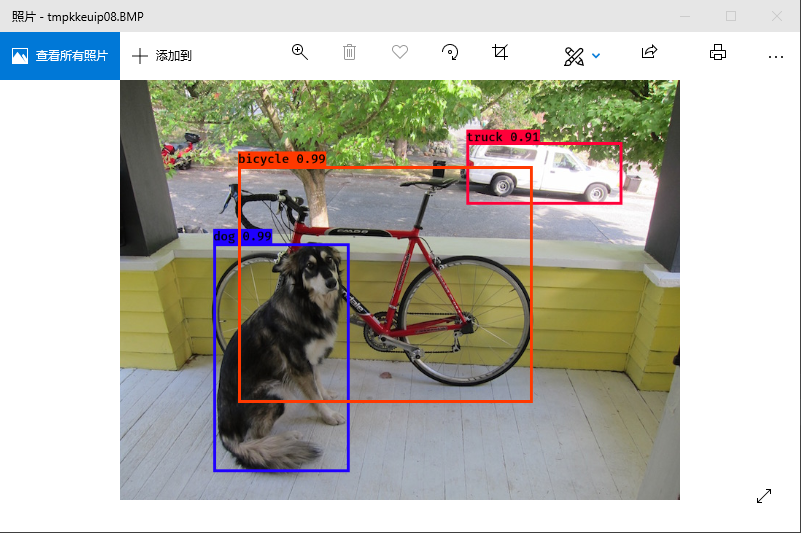


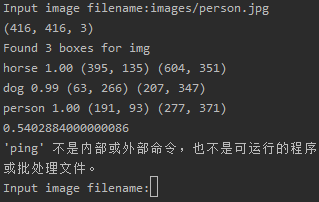
开始时比较慢，可以说很慢...

原图如下：



识别后的图片效果如下图所示：

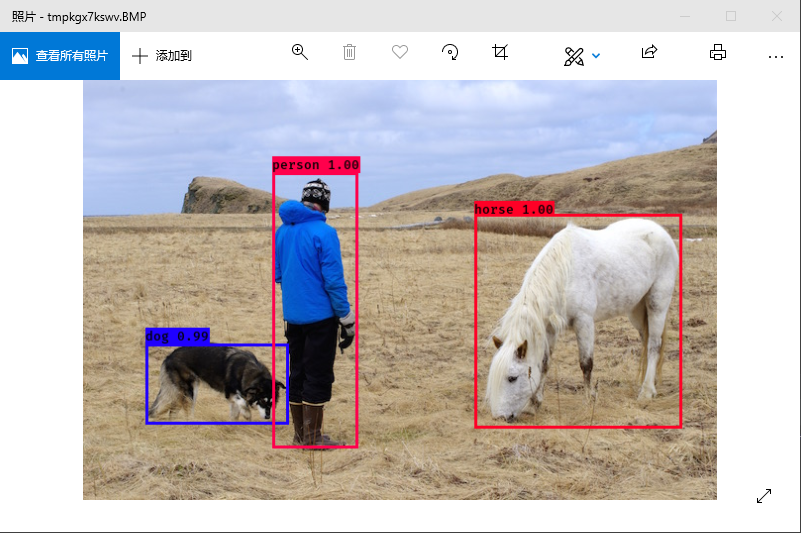




原图如下：



结果如下：

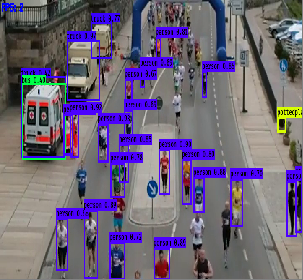


**测试视频：**

**命令行：**

**python yolo\_video.py --input images/run.mp4**

视频的截图：



比CPU版本要流畅一些，但是依然卡顿，这应该是机器的原因了，显卡OUT了吧。

其实不是，是YOLO的原因