**Keras-yolov3实验步骤**

下载KERAS-YOLO3

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

下载weights文件：

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

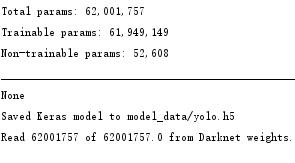
转换weights文件格式为Keras的格式：

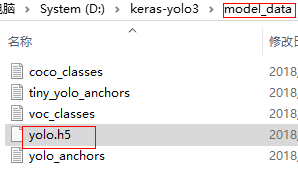
D:\keras-yolo3>C:\ProgramData\Anaconda3\envs\tensorflowgpu\python.exe d:\keras-yolo3\convert.py yolov3.cfg yolov3.weights model\_data/yolo.h5

D:\keras-yolo3>为，keras-yolov3的下载后的目录

C:\ProgramData\Anaconda3\envs\tensorflowgpu\python.exe 对应目录的python

d:\keras-yolo3\convert.py yolov3.cfg yolov3.weights model\_data/yolo.h5， 转换权重文件，并存入model\_data目录，名为yolo.h5



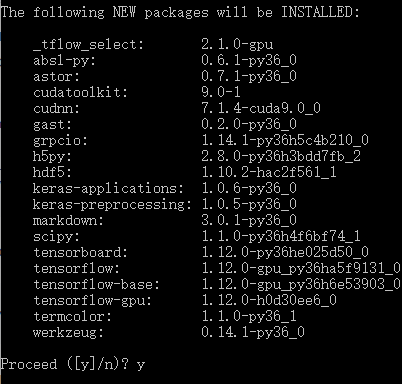


python yolo\_video.py --image --input ''

测试时报错：

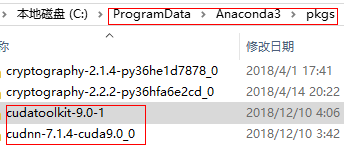
E tensorflow/stream\_executor/cuda/cuda\_dnn.cc:363] Loaded runtime CuDNN library: 7.1.4 but source was compiled with: 7.2.1. CuDNN library major and minor version needs to match or have higher minor version in case of CuDNN 7.0 or later version. If using a binary install, upgrade your CuDNN library. If building from sources, make sure the library loaded at runtime is compatible with the version specified during compile configuration.

查看搭建环境时的cudnn版本：



应该是版本低了？

查找CUDA和CUDnn的目录：



Conda 列表里windows版还没有7.1.4以上版本

还是不行哦，CONDA里没有 7.1.4以上版本

新建一个环境YOLOV3

报错：

**ImportError：DLL加载失败：找不到指定的模块。**

官方查找答案，有如下的配置：

（本机器装过8.0，9.2，10版本都不行），尝试一下如下配置再尝试一下：

<https://github.com/tensorflow/tensorflow/issues/22794>

I encountered the same issue. My configuration is:

* CUDA Toolkit v9.0 (installed without Visual Studio support)
* Tensorflow-gpu v 1.12.0 (installed with pip)
* cudnn v7.4.1.5 CUDA v9.0 compatible.
* nVidia GeForce 1070
* Windows 10 Home

After spending almost a day figuring out why tensorflow did not find the correct dll file, I found a guide on how to install cudnn that, by the way, is not as simple as it seems. The guide is written by nVidia and here you are the actions I follow.

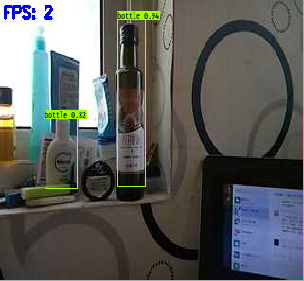
1. Download from nVidia official site, the correct cudnn version for your configuration.
2. Then extract cuda folder, it does not matter where you extract it.
3. Open File Explorer and go to the directory where you installed CUDA, in my case that was C:/Program Files/NVIDIA GPU Computing Toolkit.
4. Go to CUDA/v9.0/lib/x64. Here you have to put the file named "cudnn.lib" that you can find inside CUDA/lib/x64 (cudnn package, downloaded from nVidia)
5. Repeat the process for (left cudnn package, right CUDA installation path):

* CUDA/bin/cudnn64\_7.dll -> CUDA/v9.0/bin
* CUDA/include/cudnn.h -> CUDA/v9.0/include

1. Now you have to check if the environment variable is set correctly. Be sure CUDA\_PATH variable, with C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0 value is present, if not add it.
2. Enjoy tensorflow-gpu working on your system

测试跑通了

摄像头测试：



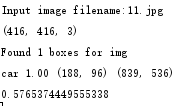
但是比较慢，视频严重延迟。

图片测试：

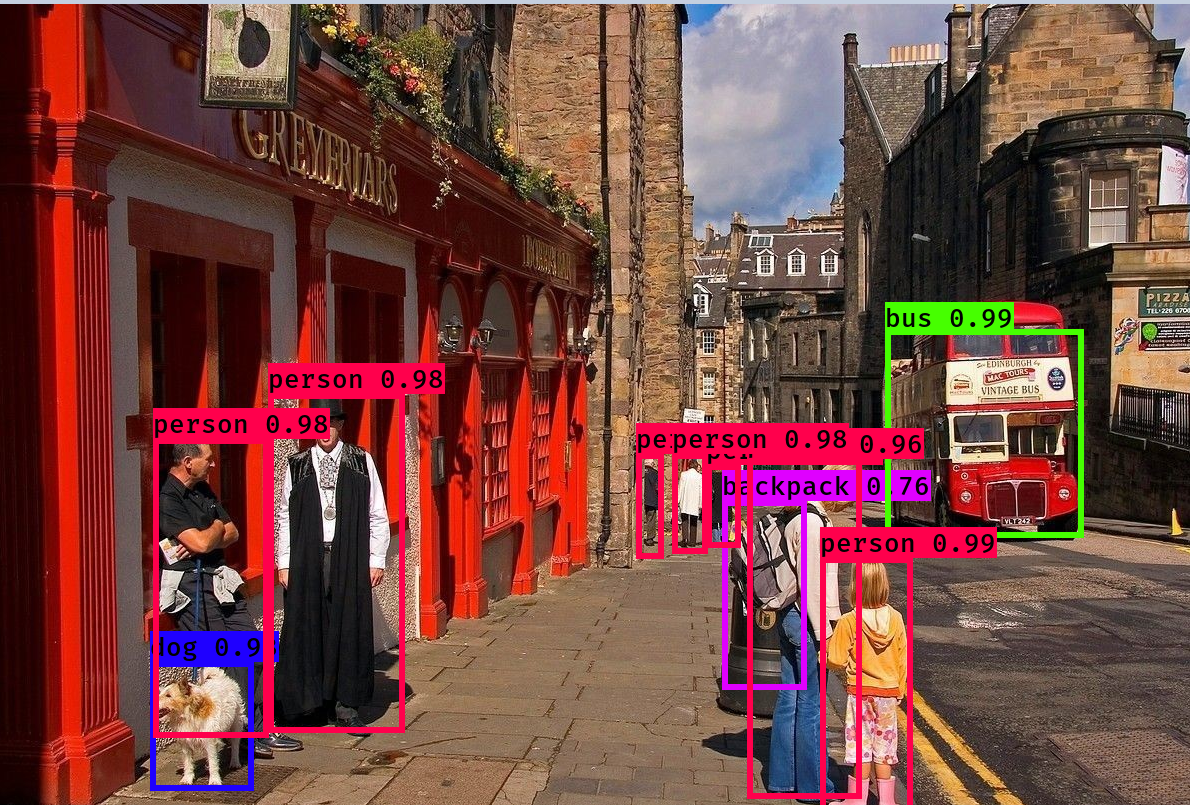
D:\keras-yolo3>C:\ProgramData\Anaconda3\envs\yolov3\python.exe yolo\_video.py --image

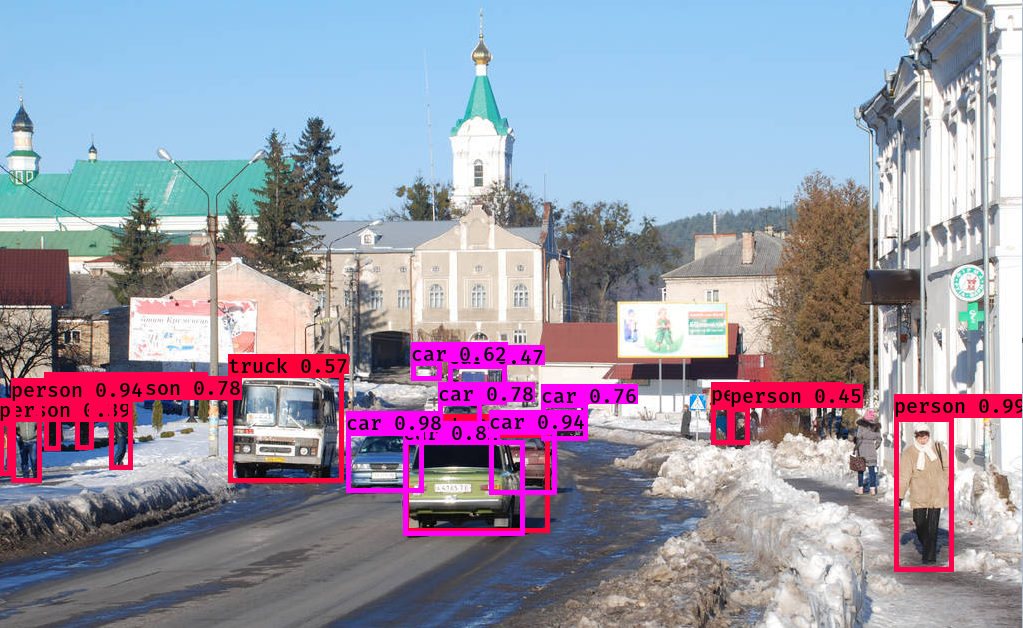
--input ''

运行后，提示输入：









**最终环境配置：**

WIN10

**CUDA Version 9.0.176**

**cudnn-9.0-windows10-x64-v7.4.1.5**

**Tensorflow-gpu 1.12.0**