# 编译darknet动态链接库

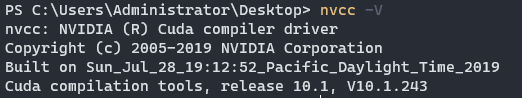
## 安装CUDA和CUDNN

安装CUDA和CUDNN并配置CUDA环境变量

在命令行输入

|  |
| --- |
| nvcc -V |

如下图显示即为安装成功



## 安装OpenCV

安装OpenCV并配置OpenCV环境变量

配置环境变量到OpenCV安装路径下的bin文件夹，如下图所示



## 克隆darknet仓库

|  |
| --- |
| git clone <https://github.com/AlexeyAB/darknet.git> |

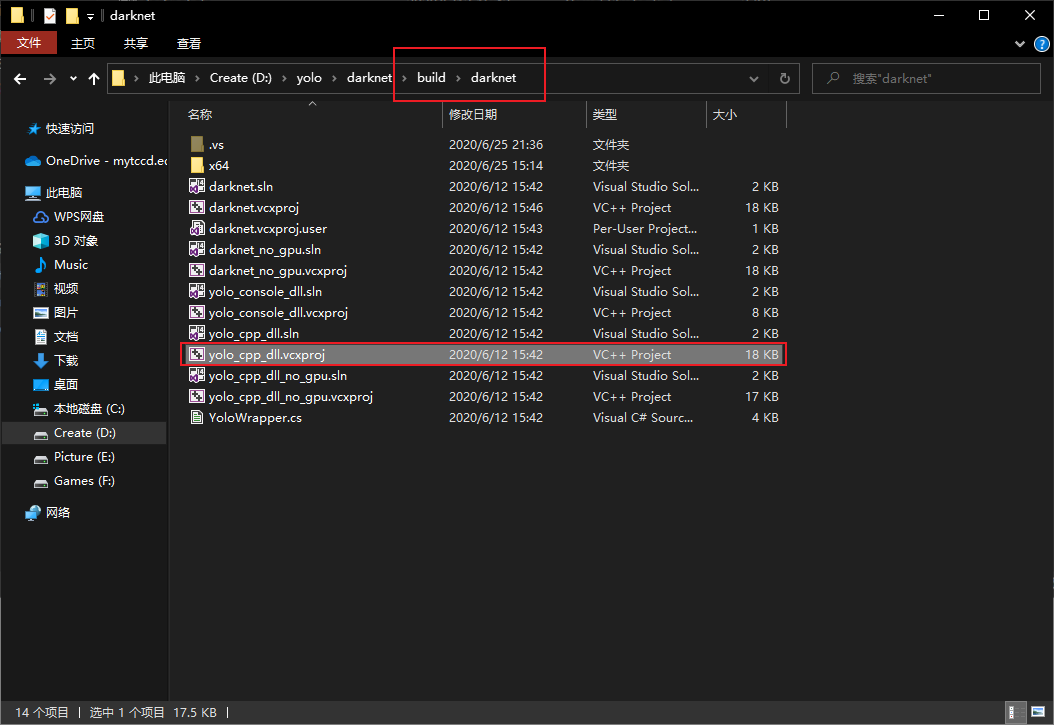
## 编译darknet动态链接库

下列方案中选择一种编译成功即可：

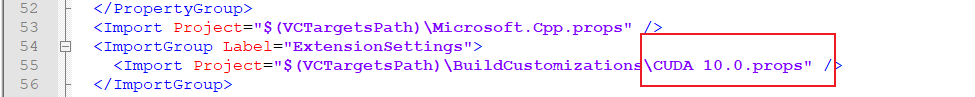
### Windows

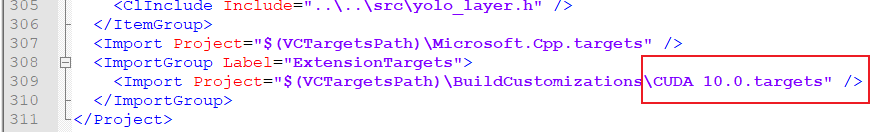
#### Visual Studio 2017

1. 进入下载好的仓库的build\darknet文件夹下，用记事本打开yolo\_cpp\_dll.vcxproj。



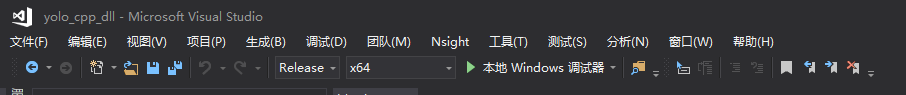
1. 把第55行和低309行的CUDA版本更改成当前电脑安装的CUDA版本。



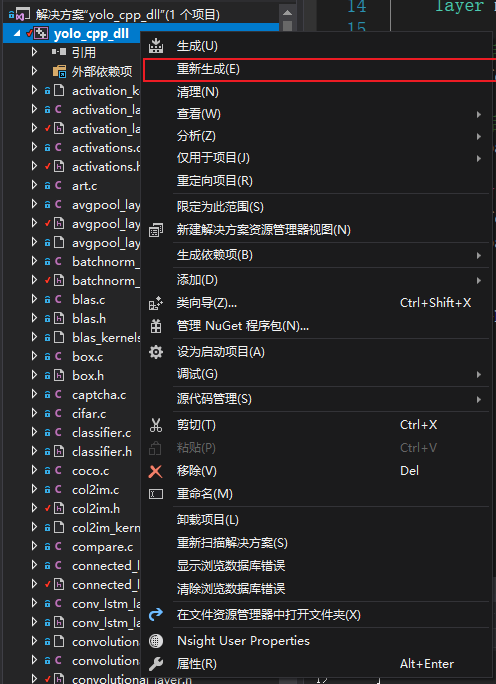


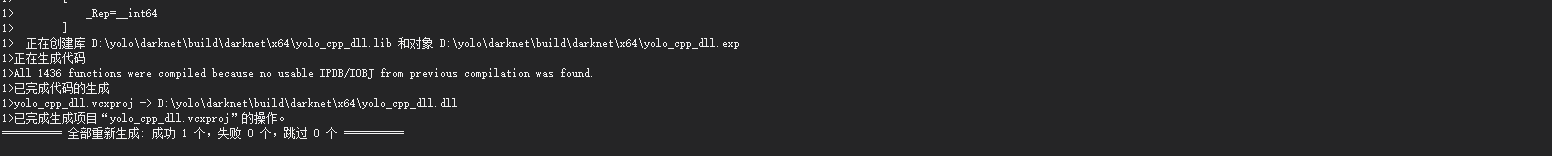
1. 使用VS2017打开yolo\_cpp\_dll.sln

解决方案配置选择Release、x64

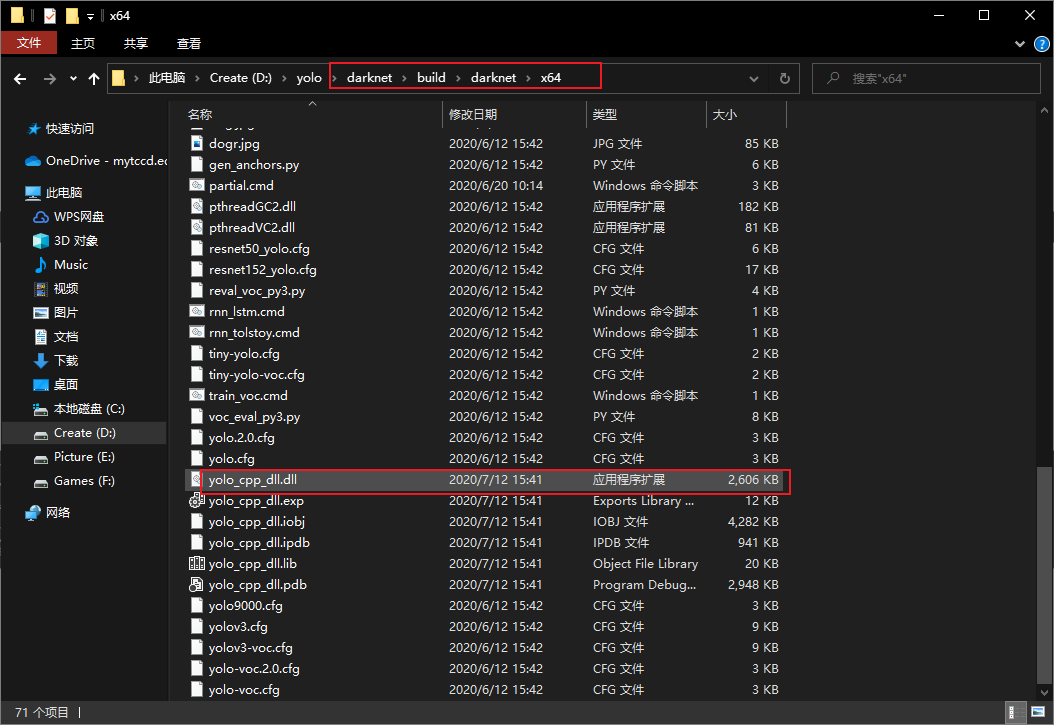


右键解决方案，选择重新生成，编译成功如下图所示：

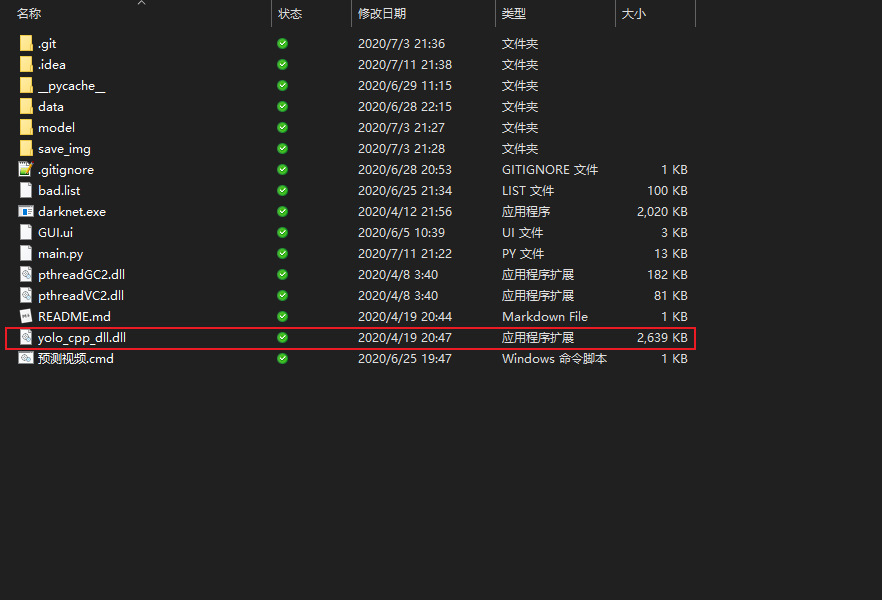




然后就可以在x64文件夹下找到yolo\_cpp\_dll.dll文件，即编译成功的dll。

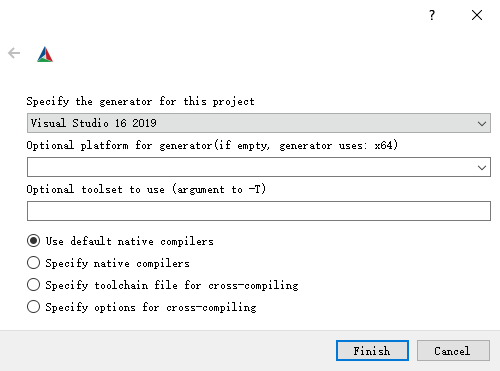


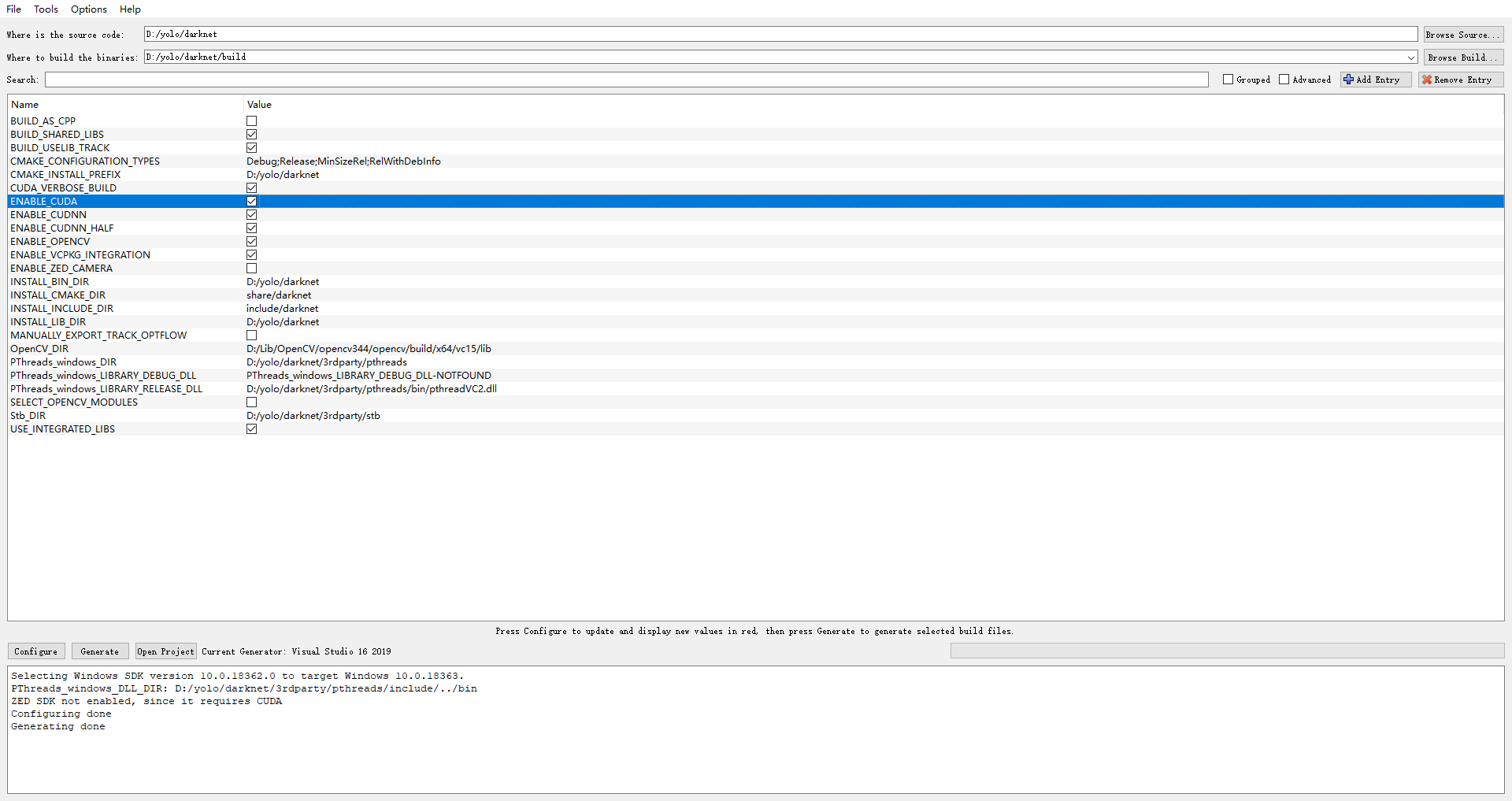
1. 使用编译成功的dll替换我们所提供的dll



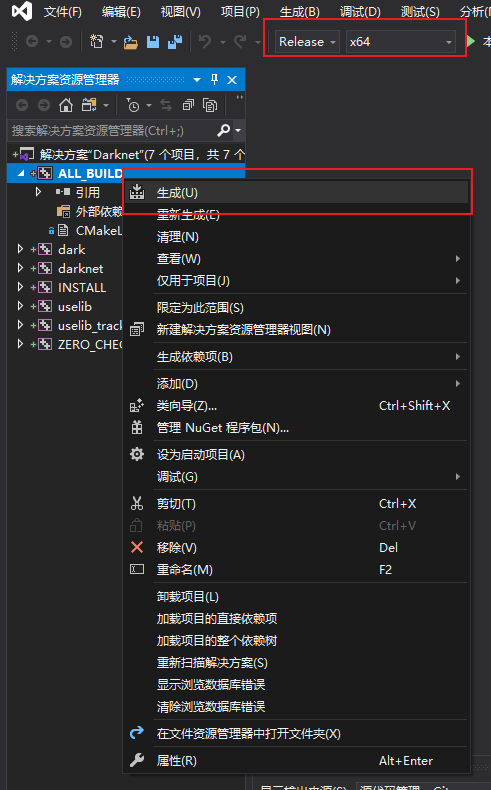
#### CMake

使用CMake-GUI打开工程选择编译路径，先Configure工程，然后选择ENABLE\_CUDA，再次Configure工程，没有问题之后Gererate生成工程。点击Open Project打开工程

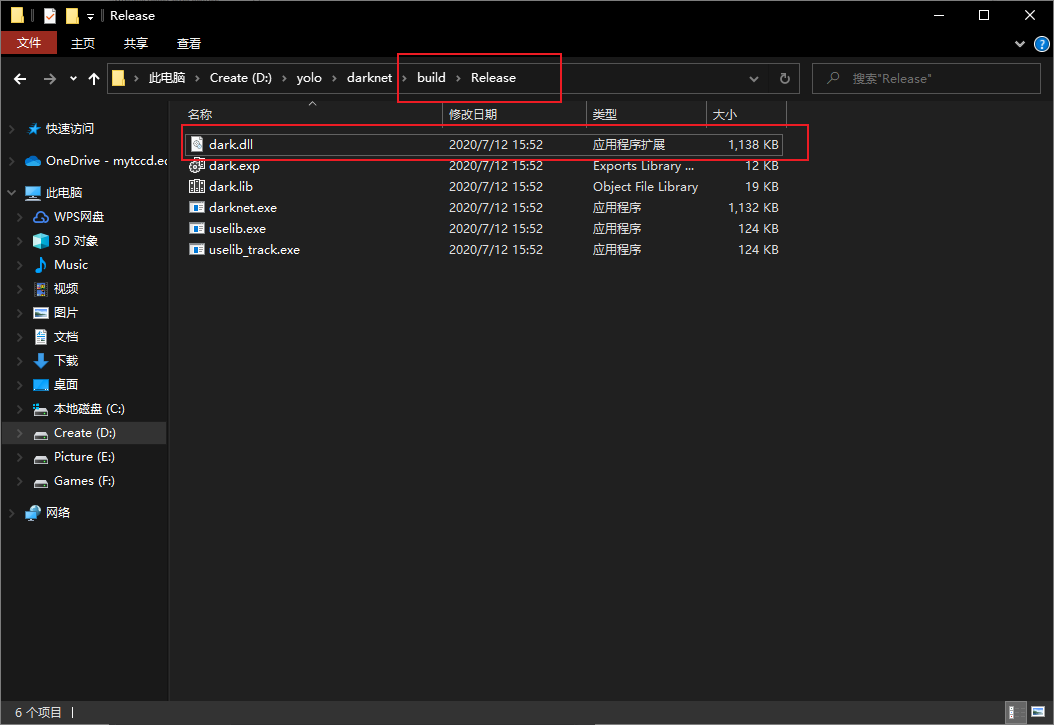




解决方案配置选择Release、x64后右键ALL\_BUILD生成工程

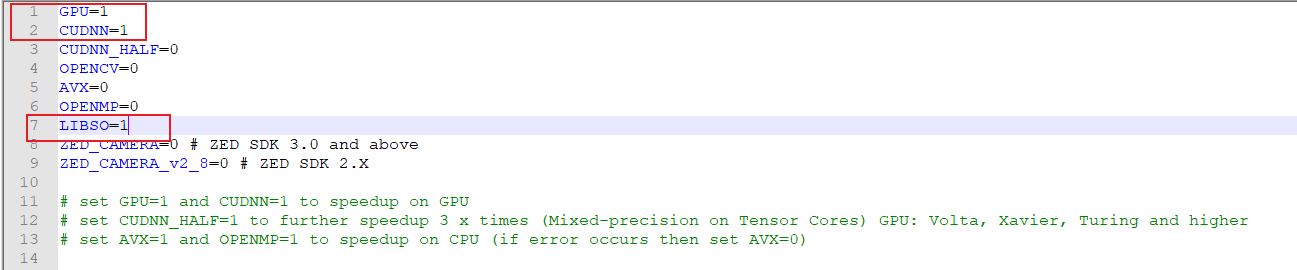


然后就可以在build/release下找到dark.dll文件，将其改名为yolo\_cpp\_dll.dll并替换我们提供的dll



### Linux

Linux更改Makefile的CPU和CUDNN都为1，LIBSO也为1后保存退出



然后执行

|  |
| --- |
| make install  make -j |

在生成目录里找到libdarknet.so文件然后拷贝到项目的根目录下。

# 导入conda环境

## 运行命令导入环境

Cd到项目根目录下的data文件夹，使用如下命令导入环境

|  |
| --- |
| conda env create -f environment.yaml |