# VS2017下编译OpenCV并使用CUDA加速

## 0 前言

### 0.1 相关版本说明

**OpenCV、CUDA建议都使用当前最新的版本，最新的版本可能就不需要想那么多事情，用老版编译可能出现一些问题，当然这仅是我个人觉得，总之，看博客看教程的时候，可以尽可能的贴近博客以及教程中的环境。**

当前环境：

OpenCV 4.5.1 （不支持VS2017以下的版本编译）

VS2017 社区版 15.9.34、Window SDK 10.0.17763.0

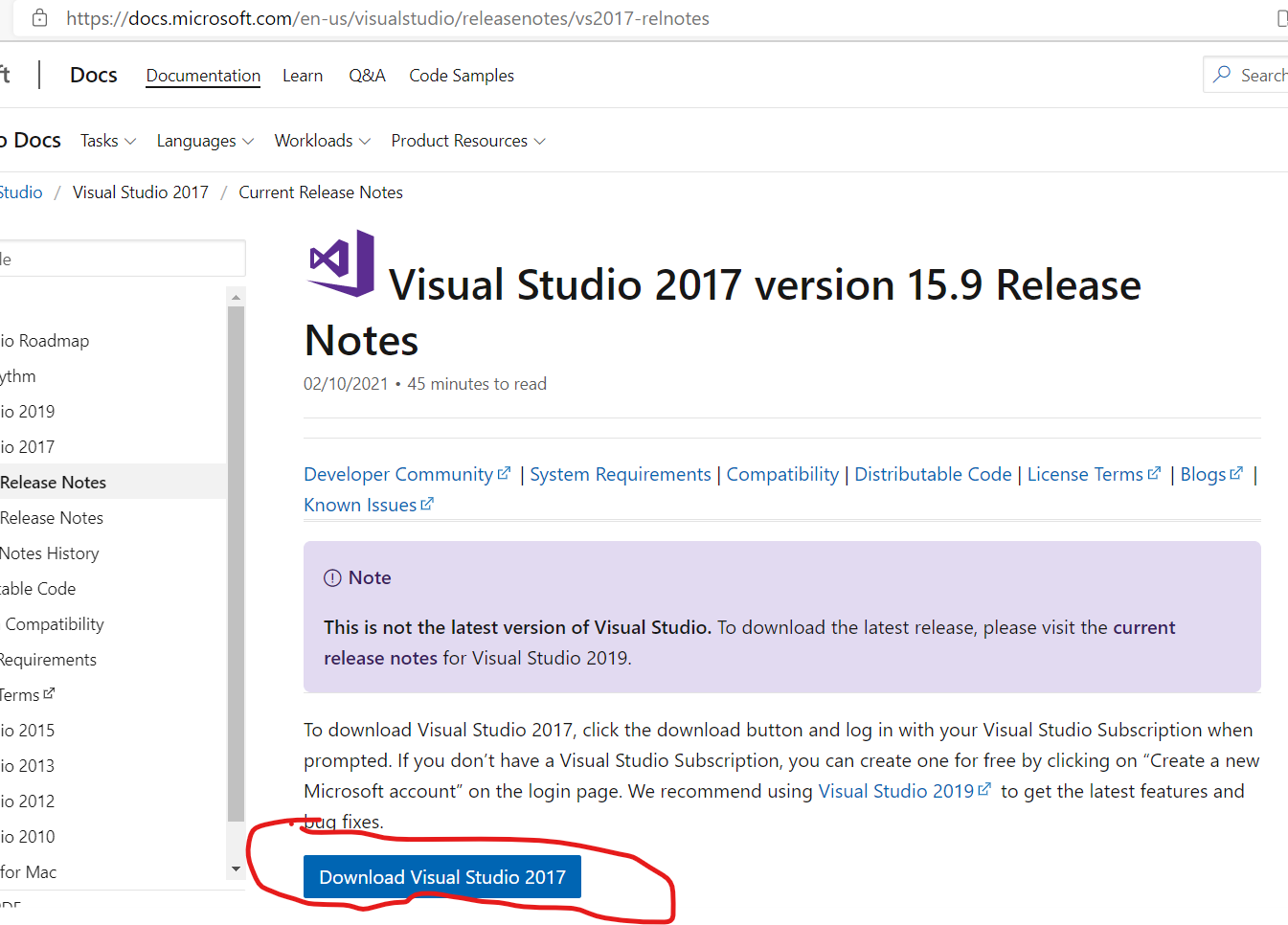
CUDA 11.2.142、CUDNN 8.1.1.33

CMake 3.20.0

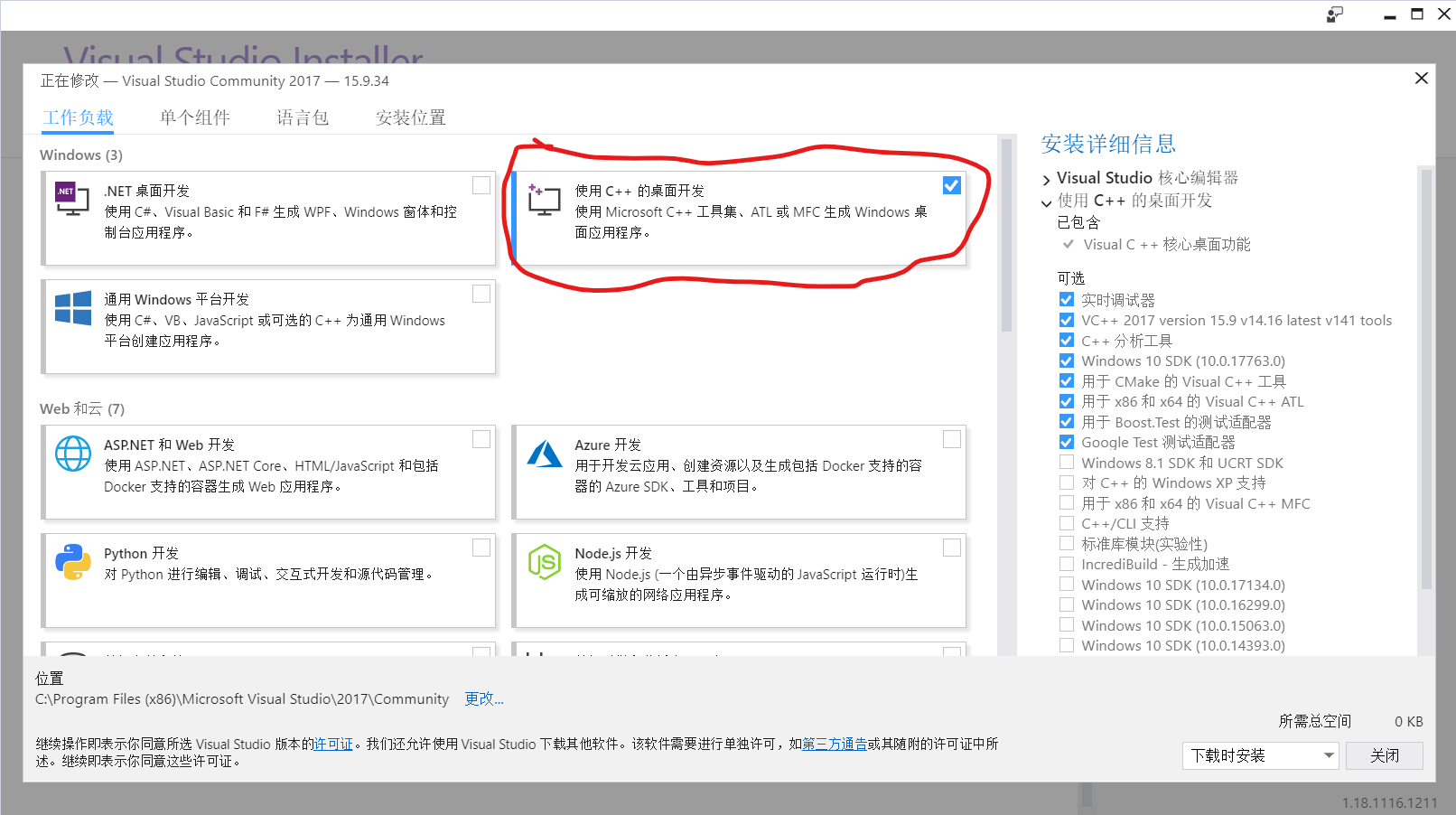
### 0.2 相关文件下载及安装

#### 0.2.1 VS2017

下载地址<https://docs.microsoft.com/en-us/visualstudio/releasenotes/vs2017-relnotes>

****

安装VS2017，根据自己的需要选择需要的开发包以及组件，如果只是OpenCV的话，点上【使用c++的桌面开发】即可。

****

#### 0.2.2 OpenCV

OpenCV下载地址

<https://opencv.org/releases/>

<https://github.com/opencv/opencv/releases>

找到合适的OpenCV版本以及OpenCV-Contrib版本，如果不需要CUDA加速以及OpenCV-Contrib中的模块的话，只需要下载OpenCV即可。可以跳过其他内容，直接看第2部分配置OpenCV的环境。

OpenCV-Contrib下载地址

<https://github.com/opencv/opencv_contrib/releases>

注意下载OpenCV对应版本的OpenCV-Contrib

#### 0.2.3 CUDA、CUDNN

CUDA <https://developer.nvidia.com/zh-cn/cuda-downloads>

CUDNN <https://developer.nvidia.com/zh-cn/cudnn>

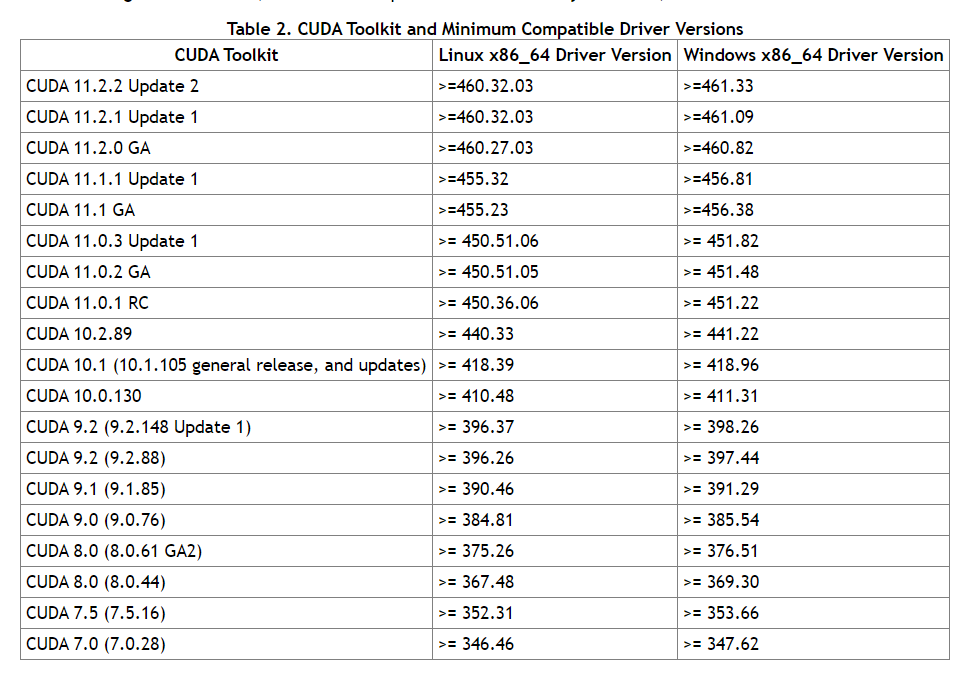
下载好CUDA，在安装的时候，可以直接选择精简，也可以自定义，只选择开发工具以及运行时，还有显卡驱动，如果显卡驱动达到了CUDA最低要求也可以不需要安装显卡驱动，推荐是安装，不需要想太多事情。可以到<https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html> 查看CUDA最低的显卡版本要求。

CUDNN 下载与CUDA对应的与系统对应的版本即可，并且CUDNN解压后只需要将bin、lib、include文件夹复制到CUDA的目录下即可。

CUDA 默认路径C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.2







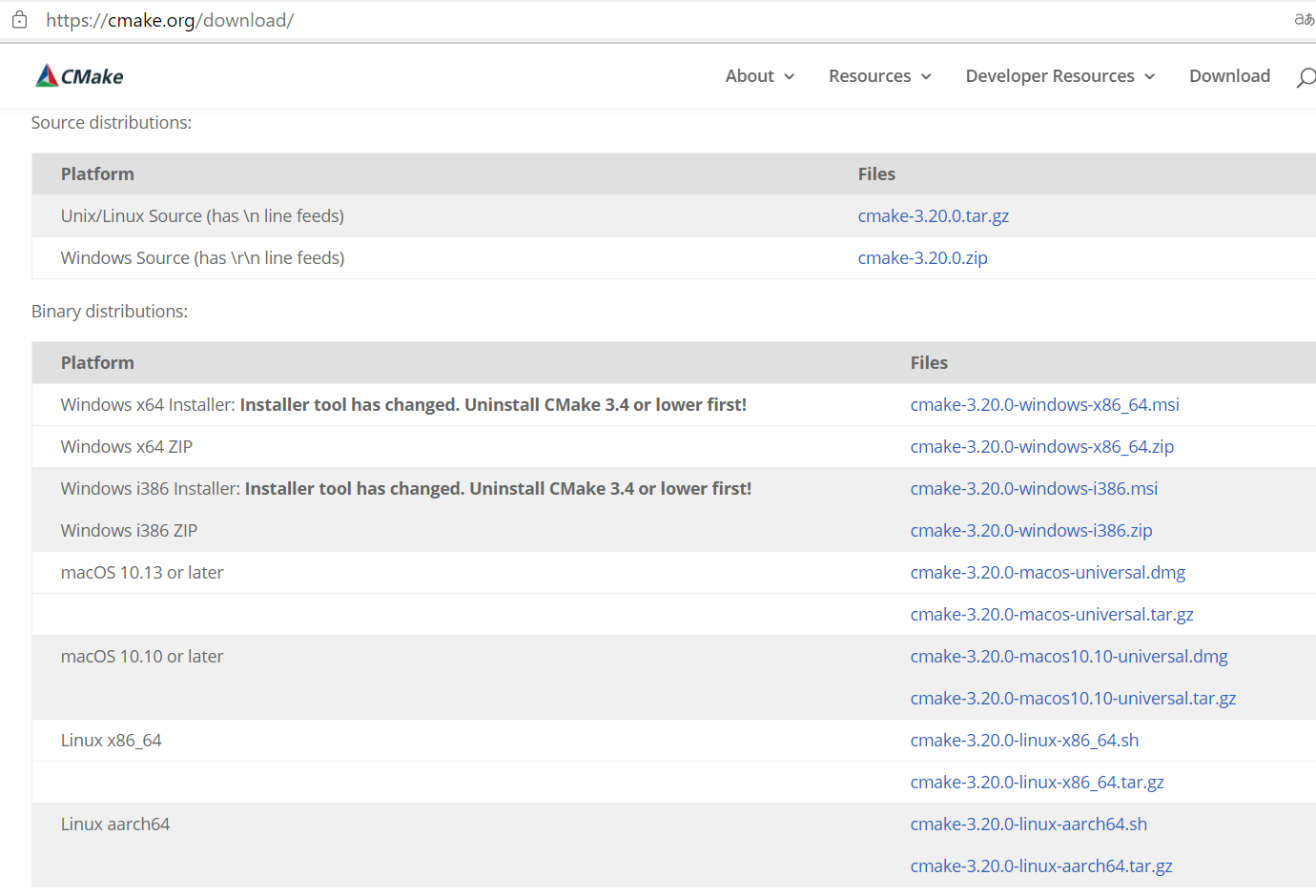
#### 0.2.4 CMake

下载地址<https://cmake.org/download/>

下载安装即可，如果下载的是安装包，解压之后还需要将bin文件夹添加到环境变量中。

添加环境变量可以看下面这个链接

https://jingyan.baidu.com/article/47a29f24610740c0142399ea.html

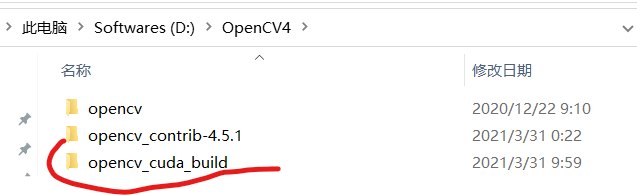


## 1 编译生成OpenCV

### CMake生成OpenCV工程

#### 1.1.1 准备工作

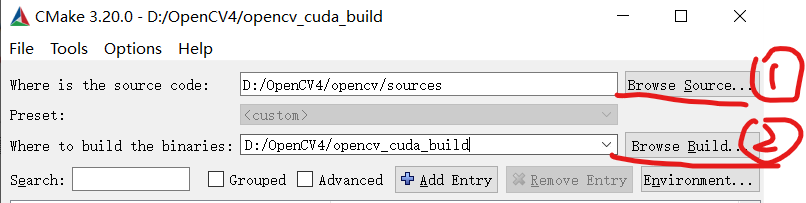
进入到opencv路径下创建一个文件夹用来存cmake生成的OpenCV的工程。如下opencv\_cuda\_build文件夹。



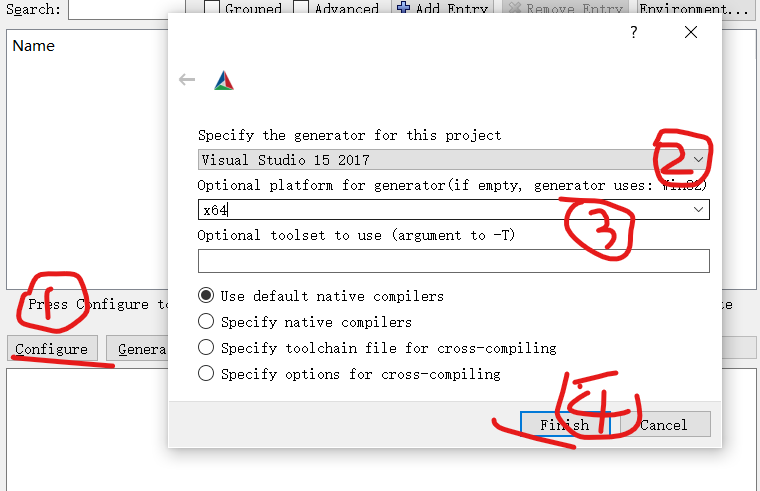
打开cmake，可以在命令行中输入cmake-gui，也可以找到cmake-gui的快捷方式，或者找到cmake所在位置的bin目录下打开cmake-gui。

选择源代码【where is the source code】在cmake-gui中点击【Browse Source】并选择OpenCV安装的路径下的sources目录。

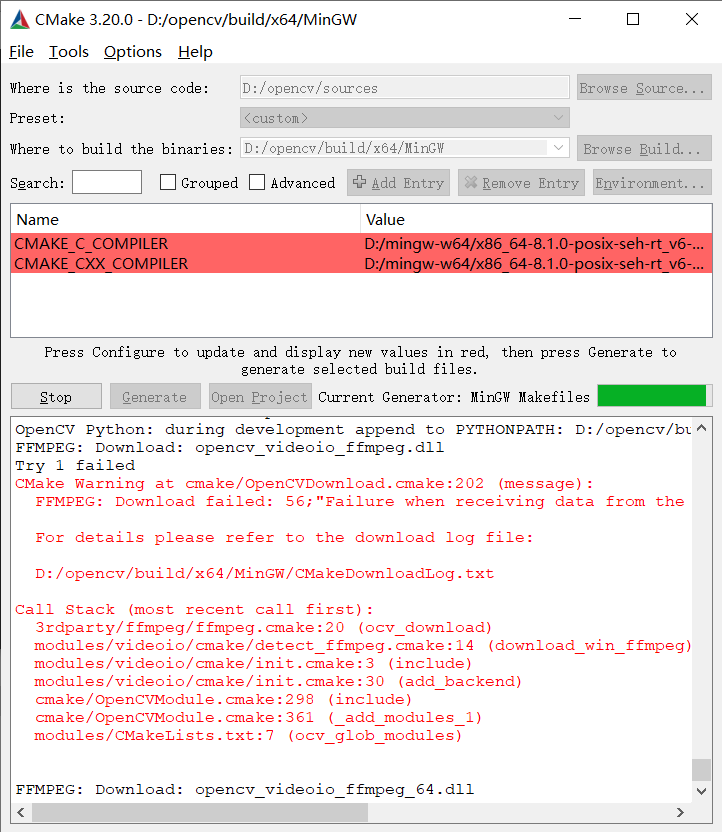
【where to build the binaries】选择上面创建好的文件夹



点击【configure】选择 VS2017，选择64位，然后点击发【finish】，它会自动开始生成工程。



#### 1.1.2 CMake 下载包出错的的处理



下面已opencv\_videoio\_ffmpeg为例。

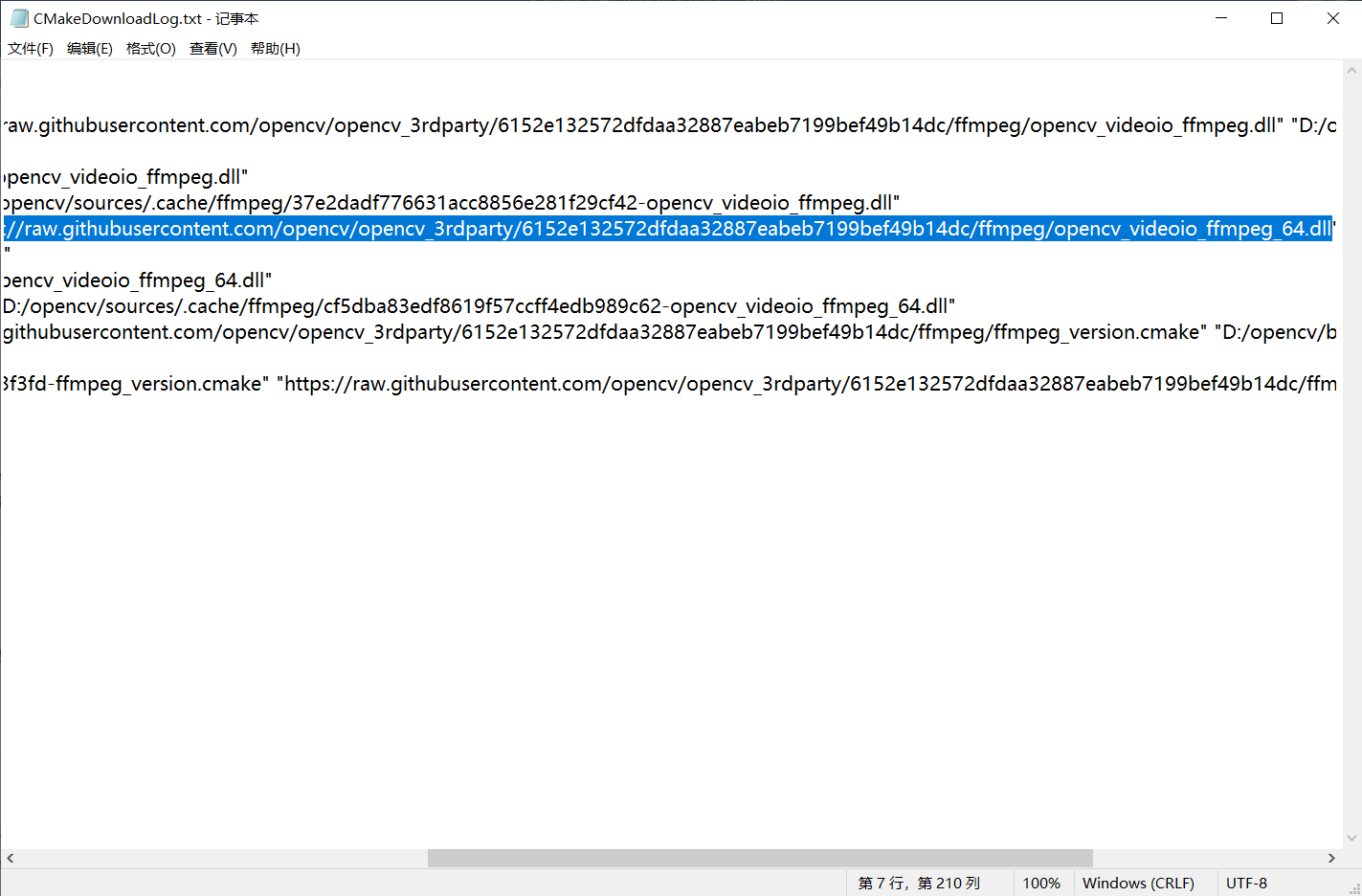
下载文件的两种方式

第一种下载方式：

找到输出编译结果文件夹（D:\OpenCV4\opencv\_cuda\_build）中的CMakeDownloadLog.txt（我这里完整的路径是D:\OpenCV4\opencv\_cuda\_build\CMakeDownloadLog.txt），复制其中github的链接输入到浏览器URL搜索栏中，下载文件，并修改后面的链接下载64位的文件。例如链接如下：

https://raw.githubusercontent.com/opencv/opencv\_3rdparty/6152e132572dfdaa32887eabeb7199bef49b14dc/ffmpeg/opencv\_videoio\_ffmpeg.dll

https://raw.githubusercontent.com/opencv/opencv\_3rdparty/6152e132572dfdaa32887eabeb7199bef49b14dc/ffmpeg/opencv\_videoio\_ffmpeg\_64.dll



第二种下载方式：

进入到OpenCV的源文件目录下的3rdparty\ffmpeg 路径下（我这里完整目录为D:\opencv\sources\3rdparty\ffmpeg）

打开目录下ffmpeg.cmake文件和ffmpeg-download.ps1.in，用ffmpeg.cmake文件中对应的FFMPEG\_BINARIES\_COMMIT值代替ffmpeg-download.ps1.in文件中url的参数名，如下图所示，如：

ffmpeg.cmake：

ocv\_update(FFMPEG\_BINARIES\_COMMIT "6152e132572dfdaa32887eabeb7199bef49b14dc")

ffmpeg-download.ps1.in：

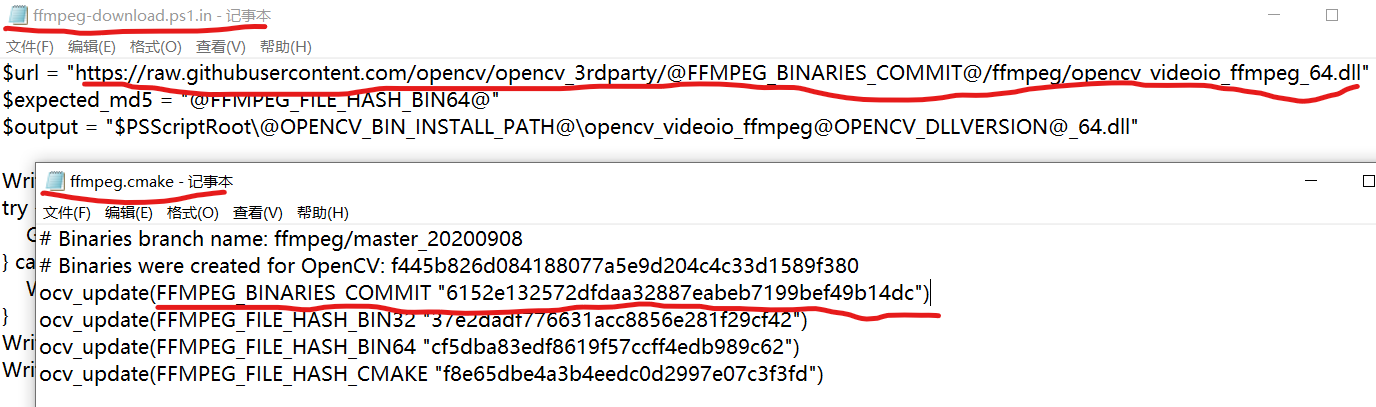
$url = "https://raw.githubusercontent.com/opencv/opencv\_3rdparty/@FFMPEG\_BINARIES\_COMMIT@/ffmpeg/opencv\_videoio\_ffmpeg\_64.dll"

最后的链接为

https://raw.githubusercontent.com/opencv/opencv\_3rdparty/6152e132572dfdaa32887eabeb7199bef49b14dc/ffmpeg/opencv\_videoio\_ffmpeg.dll

https://raw.githubusercontent.com/opencv/opencv\_3rdparty/6152e132572dfdaa32887eabeb7199bef49b14dc/ffmpeg/opencv\_videoio\_ffmpeg\_64.dll

输入到浏览器URL搜索栏中，下载文件。

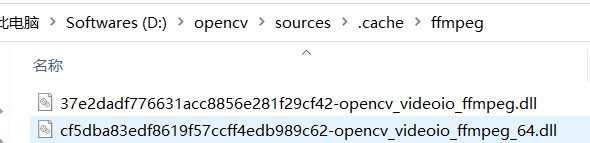


下载完成之后，根据CMakeDownloadLog.txt中给的opencv\_videoio\_ffmpeg\_64.dll文件名更改名字，并放到它指定的路径下。

将两个opencv\_videoio\_ffmpeg动态库文件放到opencv源文件的.cache\ffmpeg （我这里是在D:\OpenCV4\opencv\sources\.cache\ffmpeg）

这时候先打开任务管理器结束cmake-gui，然后再将刚刚复制过来的文件对应改名为已存在的两个未下载成功的dll文件

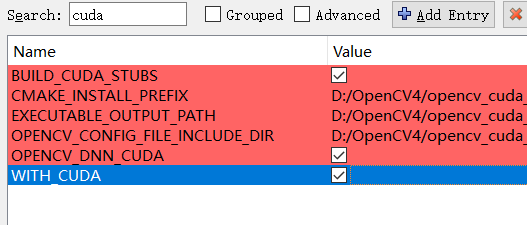
再重新启动cmake-gui点击configure，等待，直到看到Configuring done



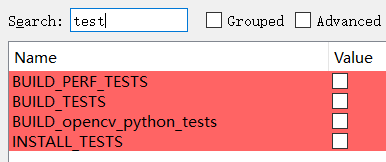
其他库文件的下载类似以上两种方法。

#### 1.1.3 勾选相关配置项

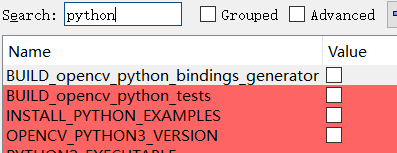
搜索CUDA，勾选WITH\_CUDA、 OPENCV\_DNN\_CUDA以及BUILD\_CUDA\_STUBS

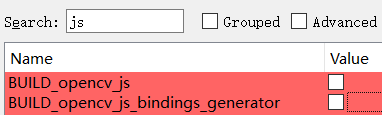


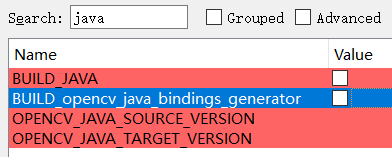
为了速度编译快一些，去掉一些无关的设置项，搜索test，去掉几个测试的模块。



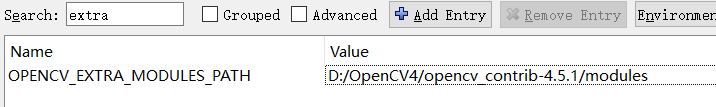
去掉python、java、javascript相关内容



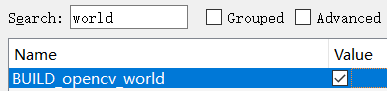




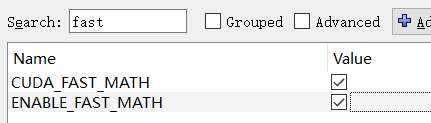
搜索extra，将opencv\_contrib的模块文件夹加入到配置项。



（可选项，建议不点），将OpenCV的库编译到一个库opencv\_world.dll



点击【configure】配置。配置完成之后，搜索fast，将如下两个点上。

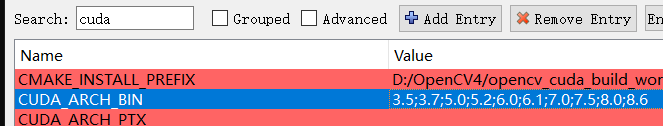


（可选，可以加快编译速度）查看本机显卡对应的gpu的算力值，

<https://developer.nvidia.com/zh-cn/cuda-gpus#collapseOne>

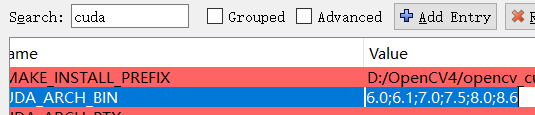
再次搜索cuda，CUDA\_ARCH\_BIN的值中留下本机算力值，去掉其他算例值

也可以将低于本机的算力值去掉

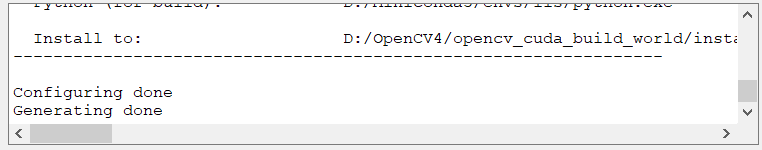


例如我这里是1050ti，在英伟达的网站上可以看到是6.1的算力，这里我将6.0以下的算力去掉。





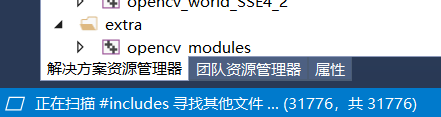
点击【configure】配置，配置完成之后点【generate】。

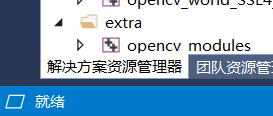
生成完成之后，再点击【open project】用VS2017打开工程。

### 编译OpenCV

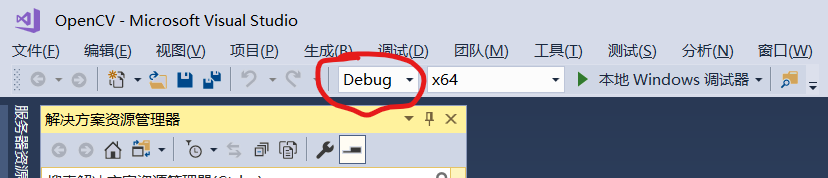
#### 1.2.1 准备工作

打开OpenCV工程之后，等待加载完成。显示【就绪】就好了。



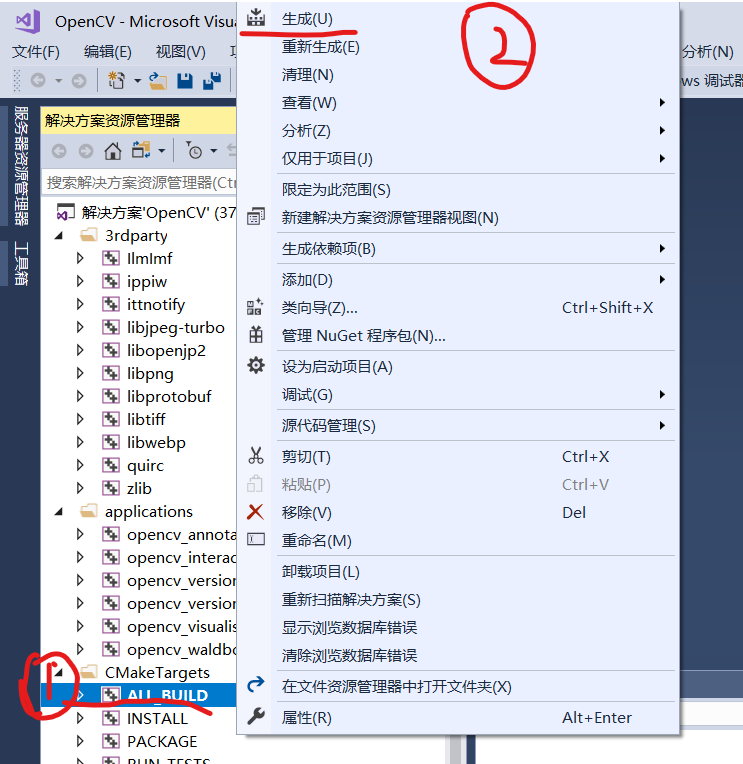


选择编译debug版本还是release版本。（建议先编译debug版本，写好代码之后先debug好，之后再编译release版本）



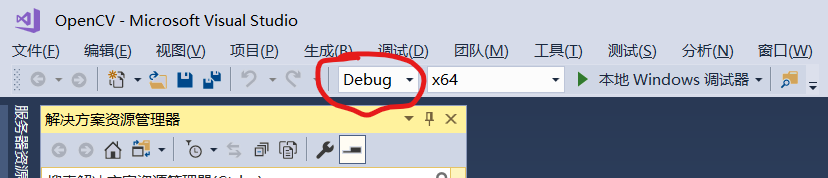
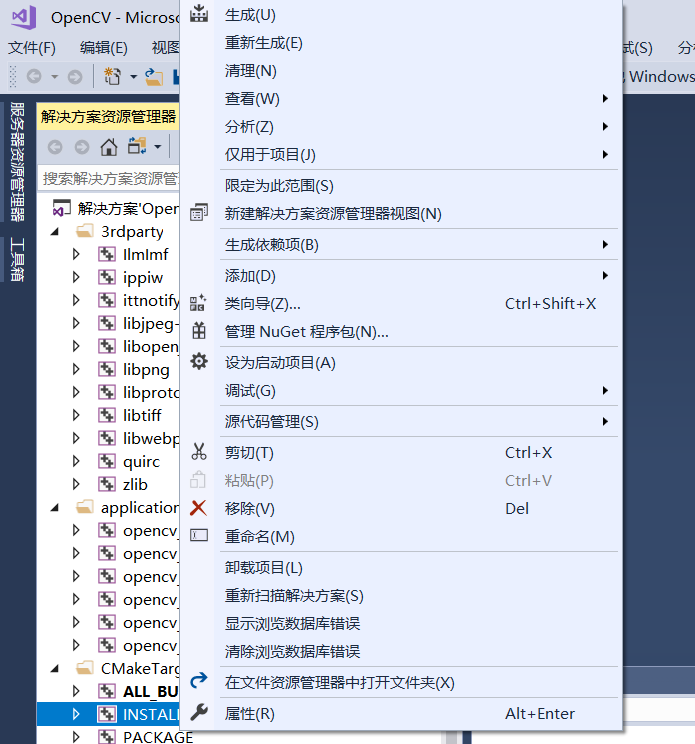
#### 1.2.3 编译生成dll库

点击ALL\_BUILD工程，点击生成，这个过程需要等待1-5个小时，根据电脑性能而定。

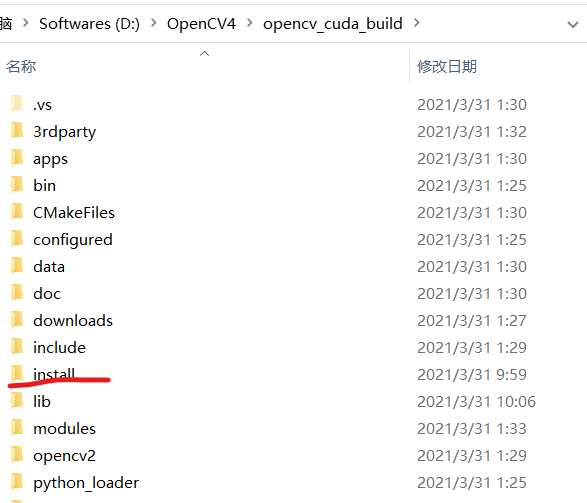


#### 1.2.4 生成install文件夹

等他编译完成后，之后，点击Install工程，让他生成install文件夹，将头文件、动态库dll等都放在一个文件夹下，这个过程不需要太久。

在自定义的存工程文件（我这里是opencv\_cuda\_build）的目录下确认一下，找到**install文件夹**，进去查看里面的内容（如下图，注：对于后边配置环境，**install文件夹**作用就像配置opencv4.5.1时的**build文件夹**一样）

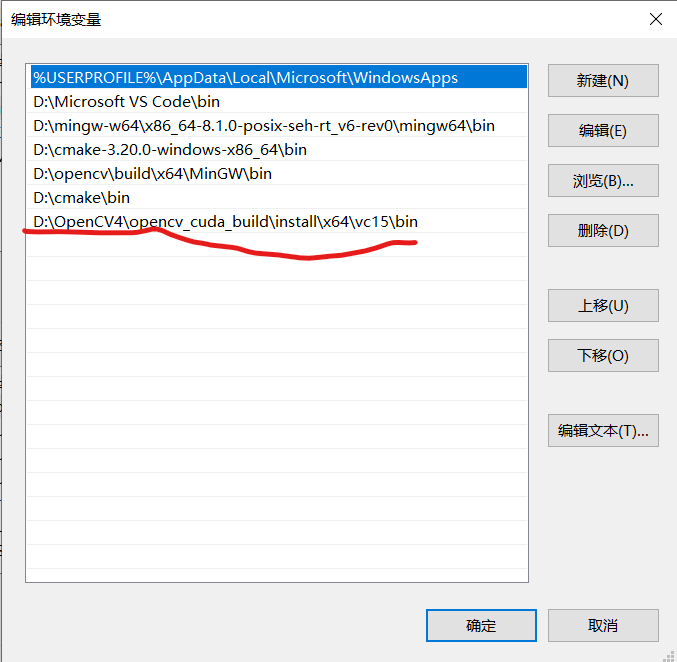


## 2 配置OpenCV环境

### 2.1 配置环境变量

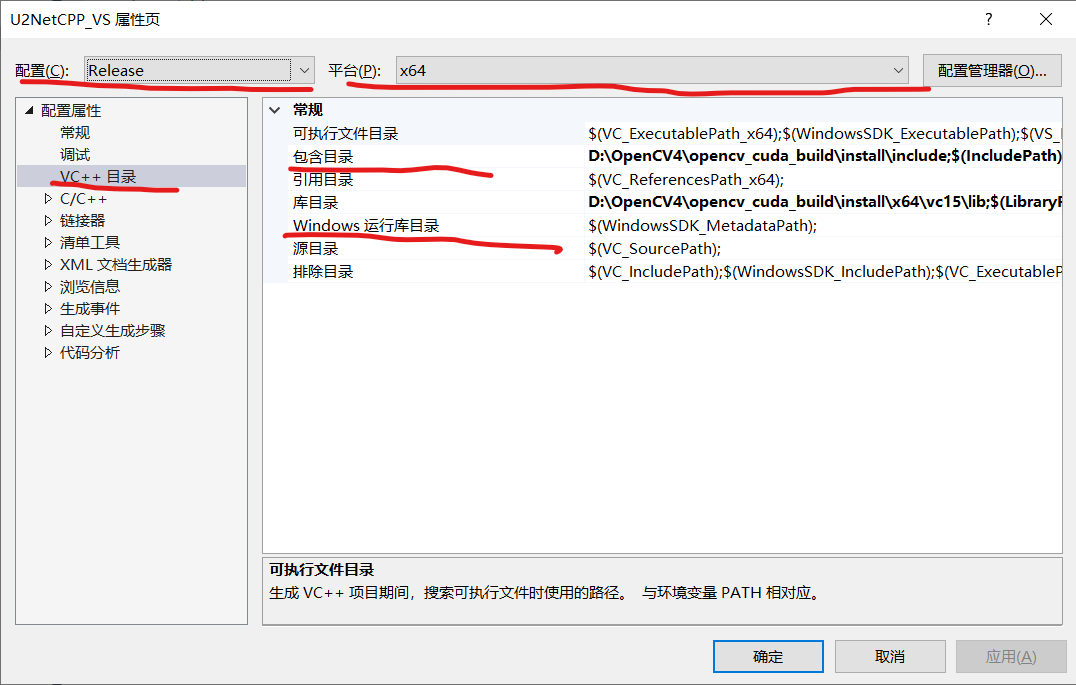
将install下的x64\vc15\bin目录配置到环境目录下。

完整路径为D:\OpenCV4\opencv\_cuda\_build\install\x64\vc15\bin

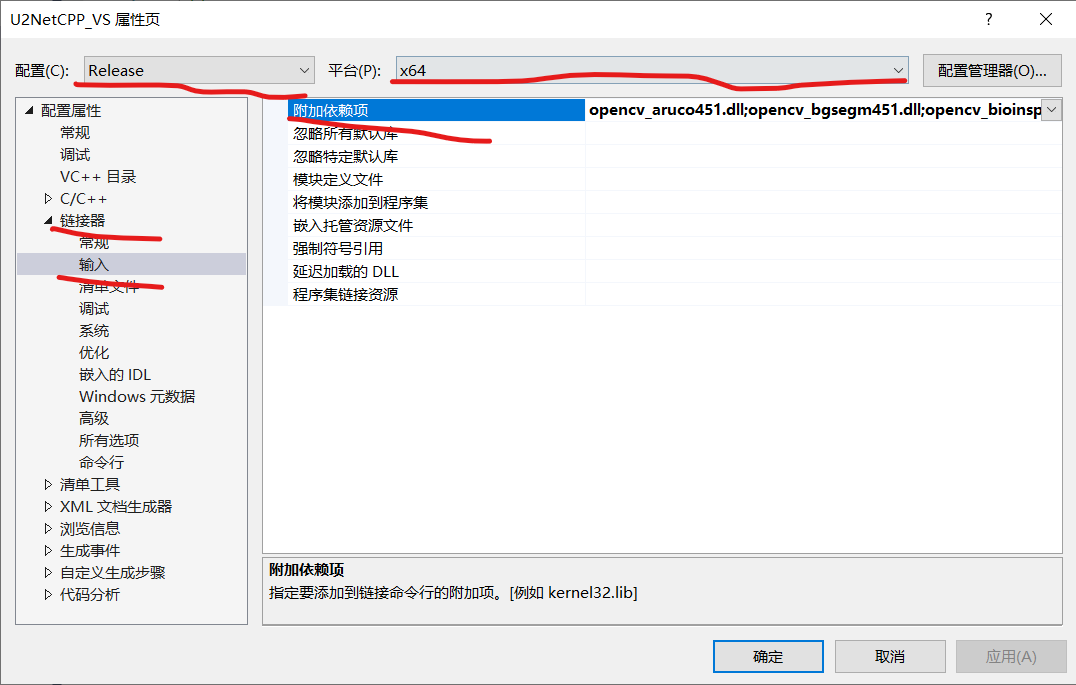


### 2.2 在VS2017中配置工程属性

创建一个项目，设置项目属性，在【C++目录】下的【包含目录】以及【库目录】中分别添加include路径以及lib路径，如下图所示



添加lib链接，在【链接器】的【输入】中的【附加依赖项】中配置lib路径下的库名称，如下图所示。可以运行如下python脚本，分别得到debug和release版本的库名。



from glob import glob

import os

lib\_dir = "D:/OpenCV4/opencv\_cuda\_build/install/x64/vc15/lib/"

lib\_version = '451'

lib\_suffix = {

    "debug" : f"\*{lib\_version}d.lib",

    "release" : f"\*{lib\_version}.lib"

}

for suffix in lib\_suffix.keys():

    print(suffix, lib\_suffix[suffix], sep="\n\n\n")

    for path in glob(lib\_dir + lib\_suffix[suffix]):

        print(os.path.split(path)[-1])

注意：配置Dubeg是只添加带d的库名，如opencv\_dnn451d.lib，而release则是不带d的，如opencv\_dnn451.lib，也就是需要配置两次，包含目录和库目录也配置两次。

另外，如何是编译到了一个lib中只需要配置opencv\_world.lib。

### 2.3 验证是否正确配置

输入以下代码，运行

#include <iostream>

#include <opencv2/opencv.hpp>

using namespace std;

using namespace cv;

int main() {

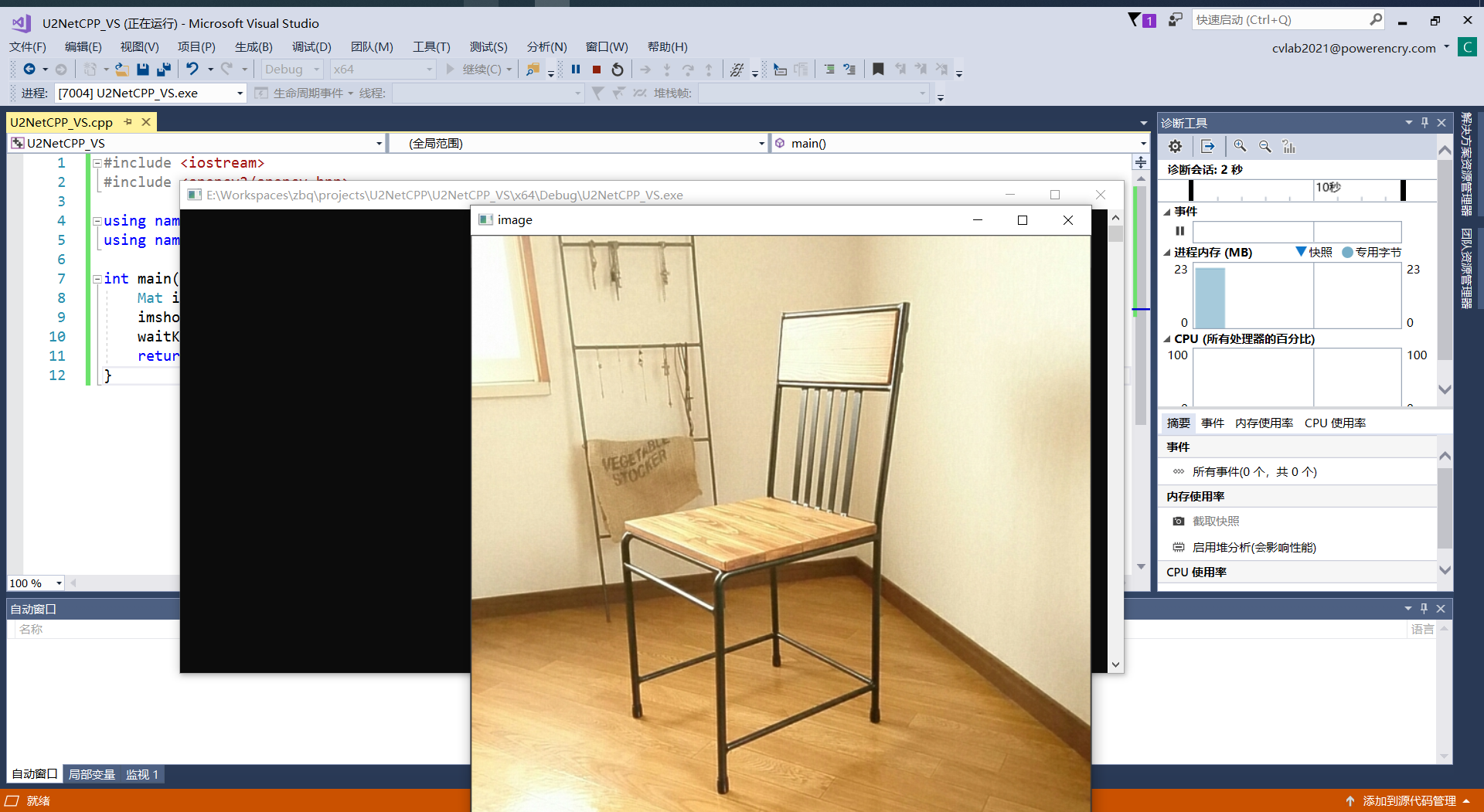
Mat img = imread("E:\\Workspaces\\zbq\\projects\\U-2-Net-master\\test\_image\\01.jpg");

imshow("image", img);

waitKey(0);

return 0;

}



## 附页

### A 参考链接

windows系统如何设置添加环境变量

<https://jingyan.baidu.com/article/47a29f24610740c0142399ea.html>

windows10+nvidia驱动+cuda10.1+cudnn安装教程

<https://blog.csdn.net/fengxinzioo/article/details/105646969>

win10+vs2017+opencv4.5.0+opencv\_contrib-4.5.0+cuda源码编译详细教程

<https://blog.csdn.net/fengxinzioo/article/details/109402921>

opencv+CUDA9.1+vs2015环境搭建，编译opencv库，调用GPU加速运算

<https://blog.csdn.net/qq_30623591/article/details/82084113>