

win 10 下 darknet配置

1 darknet本身

1. 下载软件和驱动 链接: <https://pan.baidu.com/s/19vOGqagEZeMc2Q32yZO7w> 提取码: rf48

cuda-10.0-windows10-x64-v7.6.4.38	2020/11/27 19:31	文件夹	
cmake-3.17.2-win64-x64.msi	2020/11/27 18:42	Windows Install...	24,696 KB
cuda_10.0.130_411.31_win10.exe	2020/11/27 18:50	应用程序	2,142,535...
cuda-10.0-windows10-x64-v7.6.4.3...	2020/11/27 18:44	360压缩 ZIP 文件	244,934 KB
Git-2.26.2-64-bit.exe	2020/11/27 18:43	应用程序	45,793 KB
GPU-Z.2.30.0.exe	2020/11/27 18:42	应用程序	6,929 KB
opencv-3.4.10-vc14_vc15.exe	2020/11/27 18:44	应用程序	187,055 KB
vs_Community.exe	2020/11/27 18:42	应用程序	1,362 KB

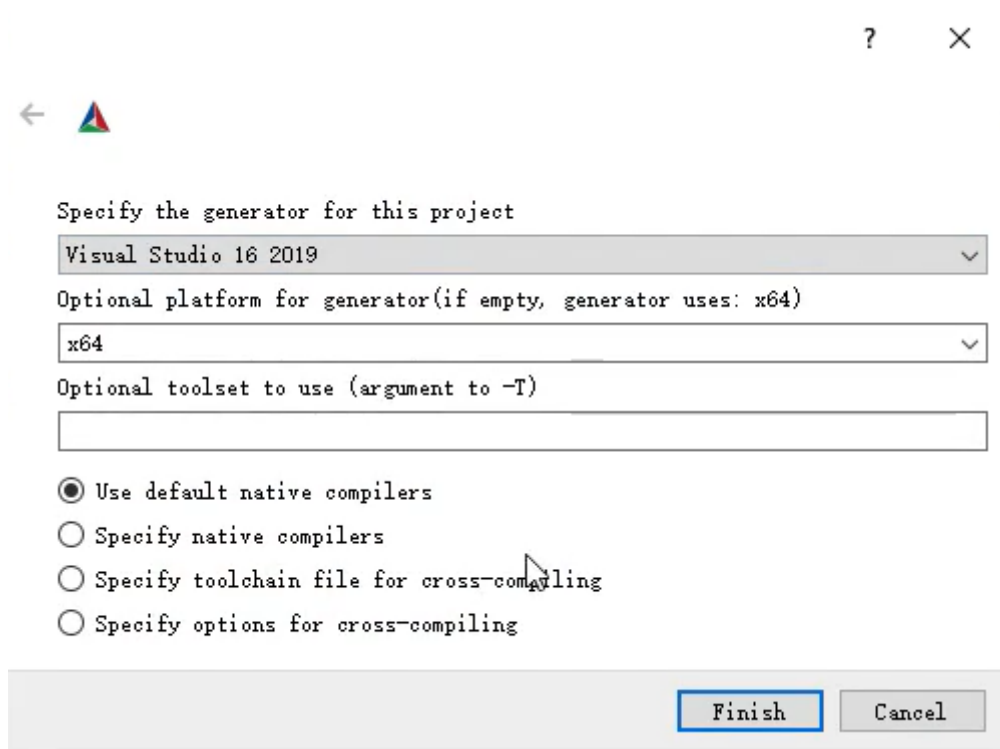
2. 添加open cv 的系统环境变量 `OPENCV_DIR` = `C:\opencv\build`

3. 双击 cuda 安装,勾选添加环境变量,然后将cudnn 粘贴到cuda下面:

> 此电脑 > Windows (C:) > Program Files > NVIDIA GPU Computing Toolkit > CUDA > v10.0

名称	修改日期	类型	大小
bin	2020/5/3 22:09	文件夹	
doc	2020/5/3 22:01	文件夹	
extras	2020/5/3 22:01	文件夹	
include	2020/5/3 22:09	文件夹	
jre	2020/5/3 22:01	文件夹	
lib	2020/5/3 22:01	文件夹	
libnvp	2020/5/3 22:01	文件夹	
nvml	2020/5/3 22:01	文件夹	
nvvm	2020/5/3 22:01	文件夹	
src	2020/5/3 22:01	文件夹	
tools	2020/5/3 22:01	文件夹	
CUDA_Toolkit_Release_Notes.txt	2018/8/26 12:31	文本文档	9 KB
EULA.txt	2018/8/26 12:31	文本文档	63 KB
NVIDIA_SL cuDNN_Support.txt	2019/9/21 11:15	文本文档	39 KB
version.txt	2018/8/26 12:32	文本文档	1 KB

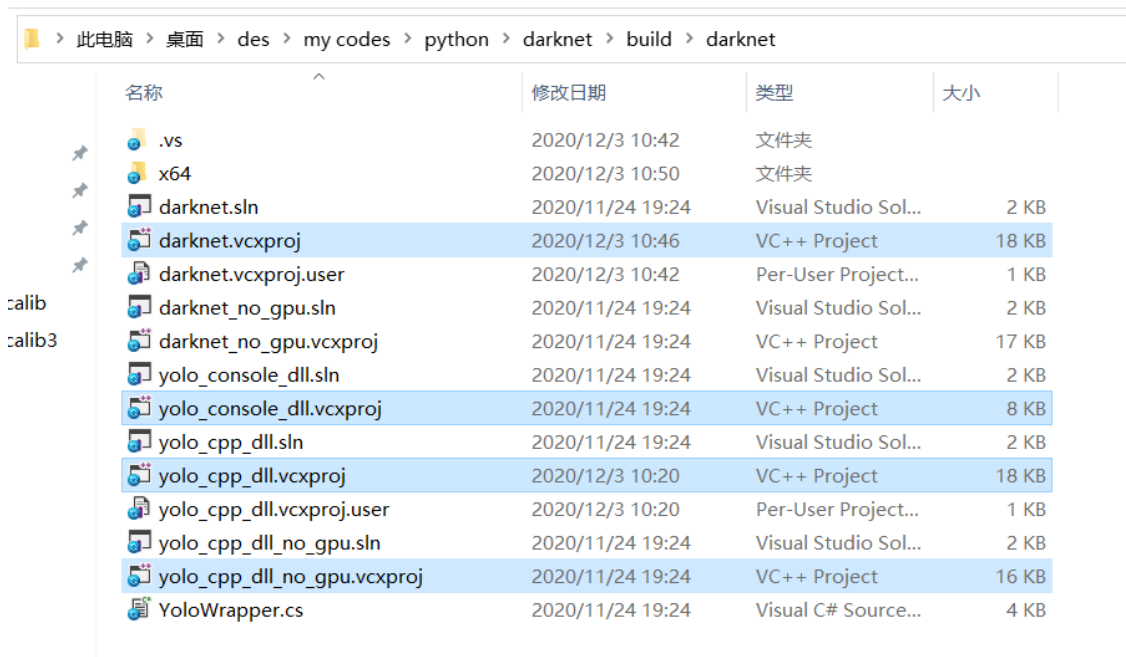
4. 从码云下载官方源码,使用cmake配置生成vs项目:



5. 生成解决方案,会打开vs,然后build,会产生.EXE文件,在对应生成的x64目录文件夹下粘贴已有的weights文件,就可以通过命令行运行了

2 使用显卡和open cv

1. vs code 打开 darknet\build\darknet 下的 project 文件,将cuda版本改为系统版本(两处,一处 in 文件末尾)



2. 用vs 打开项目,改为 x64 release 模式,添加open cv 头文件,库目录,附加包含目录,并将 cuda驱动改为相应版本(去英伟达官网查看自己的显卡对应哪个.1660ti对应 61)
3. 如果想要查看是否使用显卡可以用 nvidia-smi 同时运行darknet看显卡是否工作

问题

- 1, 可否在第一次运行生成的时候就把OpenCV和cuda依赖填好
- 2, 上图YOLO和darknet是两个都要修改吗?
- 3, 怎么以项目模式打开vs来着

3 集成qt

darknet文件依赖

此电脑 > 桌面 > des > my codes > c++ > darknet3				
	名称	修改日期	类型	大小
	model	2020/12/3 16:18	文件夹	
	CMakeLists.txt	2020/12/3 16:14	文本文档	2 KB
	CMakeLists.txt.user	2020/12/3 17:38	Per-User Project...	24 KB
	dog.jpg	2020/11/24 19:24	JPG 文件	160 KB
	main.cpp	2020/12/3 16:00	C++ Source	1 KB
calib	mainwindow.cpp	2020/12/3 16:59	C++ Source	3 KB
calib3	mainwindow.h	2020/12/3 16:03	JetBrains CLion	1 KB
	mainwindow.ui	2020/12/3 16:05	Qt UI file	2 KB
	yolo_cpp_dll.dll	2020/12/3 10:20	应用程序扩展	1,714 KB
	yolo_cpp_dll.lib	2020/12/3 10:20	Object File Library	21 KB
	yolo_v2_class.hpp	2020/11/24 19:24	C++ Header file	39 KB

两个dll 在 C:\Users\37779\Desktop\des\my codes\python\darknet\build\darknet\x64

hpp 在在 darknet\include文件夹下面

cmake 文件编写

1. 找到open cv 头文件

```
find_package(OpenCV REQUIRED)
include_directories(${OpenCV_INCLUDE_DIRS})
```

2. 动态链接

```
target_link_libraries(darknet3 PRIVATE ${OpenCV_LIBS})
target_link_libraries(darknet3 PRIVATE ${PROJECT_SOURCE_DIR}/yolo_cpp_dll.lib)
```

3. 在 add_executable 引入 yolo_v2_class.hpp

编写例子

1. 定义槽函数
2. 在release模式下运行,会在项目目录.. 生成release文件夹,内有项目.exe文件,直接运行会报错
3. 在release下建立 build 文件夹,内置:

coco.names	2020/11/24 19:24	NAMES 文件	1 KB
yolov4.cfg	2020/11/24 19:24	CFG 文件	12 KB
yolov4.weights	2020/11/27 20:16	WEIGHTS 文件	251,678 KB

4. 添加链接库依赖:

此电脑 > 桌面 > des > my codes > c++ > darknet_test > build-dark-Desktop_Qt_5_14_0_MSVC2017_64bit-Release >				
	名称	修改日期	类型	大小
	.cmake	2020/12/3 18:01	文件夹	
	CMakeFiles	2020/12/3 18:01	文件夹	
	dark_autogen	2020/12/3 18:01	文件夹	
	model	2020/12/3 18:07	文件夹	
	.ninja_deps	2020/12/3 18:01	NINJA_DEPS 文件	23 KB
-calib	.ninja_log	2020/12/3 18:07	NINJA_LOG 文件	1 KB
-calib3	build.ninja	2020/12/3 18:01	NINJA 文件	31 KB
:	cmake_install.cmake	2020/12/3 18:01	CMAKE 文件	2 KB
:3	CMakeCache.txt	2020/12/3 18:01	文本文档	15 KB
:	dark.cbp	2020/12/3 18:01	CBP 文件	22 KB
:	dark.exe	2020/12/3 18:01	应用程序	62 KB
:	dark.exe.manifest	2020/12/3 18:01	MANIFEST 文件	1 KB
	dog.jpg	2020/11/24 19:24	JPG 文件	160 KB
	opencv_world3411.dll	2020/7/17 16:36	应用程序扩展	54,953 KB
	pthreadGC2.dll	2020/11/24 19:24	应用程序扩展	182 KB
	pthreadVC2.dll	2020/11/24 19:24	应用程序扩展	81 KB
	rules.ninja	2020/12/3 18:01	NINJA 文件	3 KB
	yolo_cpp_dll.dll	2020/12/3 10:20	应用程序扩展	1,714 KB

除了open cv 的基本都在 \darknet\build\darknet\x64 文件夹下面