

前言

纯c++用户而言如果要自研产品，会一个图形界面编程工具还是有必要的。大多数c++用户，如果在Windows平台开发则多使用微软全家桶，如果是Linux平台则可能是其他工具再cmake。这篇博客将记录Windows平台，QT Creator中Opencv和Libtorch的配置。网上有较多关于使用Mingw编译Opencv源码以供QT Creator使用的，事实上，只是基于Opencv和Libtorch的api做开发的话，无需编译。正确的流程为：安装QT Creator时，勾选MSVC编译器，下载opencv的exe文件和libtorch的官方版压缩包，随后配置Release版本的项目即可。如要直接看正确步骤，前往本文“正确步骤”小节即可。各工具版本：

- QT Creator 4.11.0 (Community)
- QT 5.14.1
- OpenCV 4.5.0(官网) 4.1.1 x64(MingW轮子库版)
- Libtorch 1.7.0(官网)
- CUDA 10.2
- CUDNN(适配CUDA10.2版本)

QT Creator安装不全，编译器选择错误，OpenCv使用困难

由于自己以前主力开发工具是微软全家桶，主要Visual Studio写c++，VS code写python这样，QT Creator大致了解后简单装了下，只有MingW x64编译器，这导致OpenCV不能像在Visual Studio一样直接[官网](#)下载.exe解压添加动态库即可。由于不了解QT Creator，配置时直接按照网上部分博客在.pro中修改，导致出错。

直接下载opencv官网.exe，使用MingW编译主要错误表现为：

1. 控制台无法输出，即QT Creator最下方3 应用程序输出只有Debugging starts和Debugging has finished字样。
2. cv::Mat可以使用，但是imread和imshow均无法使用，右键项目重新构建时出现 undefined reference to `cv::imread(cv::String const....

错误undefined reference to也就是“未定义的标识符”错误，这是因为没有正确关联.lib文件，工程只能找到函数声明，无法找到定义。然后一顿折腾，以为自己.pro文件中的INCLUDEPATH和LIBS配置不对，各种魔改都没有效果。最后看某个[博客](#)说因为OpenCV官网提供的是MSVC编译的，如果想基于MingW开发，需要前往[这个GitHub](#)上下载需要版本。OpenCV在2.4.10及之后版本不再提供MingW编译的Lib。

如果要跨平台做基于OpenCV的开发，要么自己用MingW编译一次OpenCV源码，要么用网上已有的轮子。

选用[轮子库](#)里的MinGW-OpenCV-4.1.1-x64，在.pro文件中配置好库文件如下：

```
INCLUDEPATH += your path to\MinGW-OpenCV-4.1.1-x64\include

LIBS+=your path to\MinGW-OpenCV-4.1.1-x64\x64\mingw\lib\libopencv_*.a
```

*是正则表示任意字符串。QT Creator中的配置路径很有趣，同时接受/, //, \和\。在.pro文件改好后，选择构建build，执行qmake，然后再右键项目，选择重新构建，此时无论项目配置使用Release或者Debug编译均可通过。同时，也可以在Linux中做OpenCV开发。

Libtorch只能用MSVC编译器直接调用

有了配置成功OpenCV的经验，就自然想继续用MingW配置Libtorch，找到pytorch官网提供的[libtorch](#)下载Debug或者Release版本。以Release版本为例配置.pro如下：

```
INCLUDEPATH += your path to\MinGW-OpenCV-4.1.1-x64\include \
your path to\libtorch17release\include \
your path to\libtorch17release\include\torch\csrc\api\include

LIBS+=your path to\MinGW-OpenCV-4.1.1-x64\x64\mingw\lib\libopencv_*.a \
your path to\libtorch17release\lib\*.lib \
your path to\libtorch17release\lib\*.dll
```

qmake, 重新构建，产生159个warning，12个error。警告不用管，项目配置Debug或者Release编译，错误都有有：

1. include\c10\macros\Macros.h:287: error: '__assert_fail' was not declared in this scope
__assert_fail
2. include\ATen\core\ivalue_inl.h:719: error: expected unqualified-id before '(' token
TORCH_CHECK(obj->slots().size() == 1,

重装QT Creator，配置MSVC编译器

可以在QT安装路径中使用QT manager补装，但是比较麻烦。直接在控制面版中卸载QT Creator，[官网](#)下载或者搜索其他网址下载，官网加载很慢可以搜索下载。

参考[QT Creator安装](#)，重装时，在选择组件部分除了默认的Developer and Designer Tools中的MingW和CDB，还要勾选MSVC部分。

安装好MSVC编译器，还要配置，配置时需要cdb.exe。选择工具->选项->构建套件(Kits)->MSVC 2017 x64->调式器，添加cdb.exe。如果没有，[下载安装](#)即可。装好后在添加cdb调试器。

重新用MSVC 2017编译器，配置OpenCV和Libtorch

配置好MSVC编译器后，在.pro文件加入配置如下：

```
INCLUDEPATH += your path to\opencv-4.5.0-vc14_vc15\opencv\build\include \
your path to\libtorch17debug\include \
your path to\libtorch17debug\include\torch\csrc\api\include

LIBS+=your path to\opencv-4.5.0-vc14_vc15\opencv\build\x64\vc15\lib\opencv_world450d.lib \
your path to\libtorch17debug\lib\*.lib
```

执行qmake，重新构建。发现报错主要来自IValue.h和IValue_init.h文件，显示7个错误

- 语法错误: 标识符“IValue”
- 意外的标记位于“;”之前
- “c10::IValue”:非法的成员初始化
- “Tag”: 不是类或命名空间名称
- 语法错误，缺少“;”等。

解决方法参考[引用3](#)，分两步

1. 在引用 `#include <torch/torch.h>` 的地方，这样写

```
#undef slots
#include <torch/torch.h>
#define slots Q_SLOTS
```

2. 在报错的D:\libtorch\include\ATen\core\ivalue.h头文件和IValue_init.h文件中找出如下3行内容，注释掉。共有多处

```
/// \cond DOXYGEN_CANNOT_HANDLE_CONSTRUCTORS_WITH_MACROS_SO_EXCLUDE_THIS_LINE_FROM_DOXYGEN
C10_DEPRECATED_MESSAGE("IValues based on std::vector<T> are potentially slow and deprecated. Please use torch::Tensor instead.")
/// \endcond
```

至此，重新构建不再出错。但是运行报错：**cdb process terminated**。排查错误来源，发现只使用OpenCV是没有问题的，但是加入libtorch则出错。没办法，只能将项目配置Debug转成Release。或者运行也不报错，但是当运行代码中有libtorch相关的模型操作，无论加载或者forward，均会终止运行。

正确步骤

至此，坑已经全部踩完。CUDA和CUDNN的安装网上方法很多，就不赘述。安装好后，QT完整的深度学习配置步骤为：

1. 下载安装QT Creator，配置好环境。[官网](#)下载或者搜索其他网址下载，官网加载很慢可以搜索下载。注意安装时勾选MSVC编译器组件。项目选用MSVC编译，在工具->选项->构建套件(Kits)->MSVC 2017 x64配置MSVC 2017 x64，选择c和c++编译器为amd64。
2. 如果电脑没有cdb.exe文件(Everything搜查)，则[下载安装](#)，安装完毕后选择工具->选项->构建套件(Kits)->MSVC 2017 x64->Debugger(调式器)，添加cdb.exe。
3. 下载[OpenCV](#)和[libtorch](#)。将正确路径配置到项目的.pro文件中，在.pro文件末尾添加

```
INCLUDEPATH += your path to\opencv-4.5.0-vc14_vc15\opencv\build\include \
your path to\libtorch17release\include \
your path to\libtorch17release\include\torch\csrc\api\include
```

```
LIBS += -Lyour path to\opencv-4.5.0-vc14_vc15\opencv\build\x64\vc15\lib -lopencv_world450 \
-Lyour path to\libtorch17release\lib -lc10 -ltorch -lc10_cuda -lcaffe2_detectron_ops_gpu -lc10d -ltorch_cpu \
-ltorch_cuda -lgloo -lcaffe2_module_test_dynamic -lasn1 -lcaffe2_nvrtc -lclog -lcpuinfo -ldnnl -lfbgemm -lgloo_
-lmkldnn -INCLUDE:?warp_size@cuda@at@@YAHXZ
```

将your path to替换为你本地路径。

4. 项目配置为Release模式，运行qmake成功，右键项目重新构建。可能产生错误有：

语法错误: 标识符“lValue”;

意外的标记位于“;”之前;

“c10::lValue”:非法的成员初始化;

“Tag”: 不是类或命名空间名称;

语法错误, 缺少“;”。

解决方法参考[引用3](#)，在引用 `#include <torch/torch.h>` 的地方，这样写

```
#undef slots
#include <torch/torch.h>
#define slots Q_SLOTS
```

在报错的D:\libtorch\include\ATen\core\ivalue.h头文件和lValue_init.h文件中找出如下3行内容，注释掉。共有多处

```
/// \cond DOXYGEN_CANNOT_HANDLE_CONSTRUCTORS_WITH_MACROS_SO_EXCLUDE_THIS_LINE_FROM_DOXYGEN
C10_DEPRECATED_MESSAGE("IValues based on std::vector<T> are potentially slow and deprecated. Ple
/// \endcond
```

至此，重新构建不再出错

5. 成功运行，测试的main.cpp文件代码如下：

```

#include "mainwindow.h"
#include<opencv2/opencv.hpp>
#include <QApplication>
#include<iostream>
#undef slots
#include<torch/script.h>
#include<torch/torch.h>
#define slots Q_SLOTS

class ConvReluBnImpl : public torch::nn::Module {
public:
    ConvReluBnImpl(int input_channel=3, int output_channel=64, int kernel_size = 3);
    torch::Tensor forward(torch::Tensor x);
private:
    // Declare layers
    torch::nn::Conv2d conv{ nullptr };
    torch::nn::BatchNorm2d bn{ nullptr };
};
TORCH_MODULE(ConvReluBn);

ConvReluBnImpl::ConvReluBnImpl(int input_channel, int output_channel, int kernel_size) {
    conv = register_module("conv", torch::nn::Conv2d(torch::nn::Conv2dOptions(input_channel, out
    bn = register_module("bn", torch::nn::BatchNorm2d(output_channel));
}

torch::Tensor ConvReluBnImpl::forward(torch::Tensor x) {
    x = torch::relu(conv->forward(x));
    x = bn(x);
    return x;
}

int main(int argc, char *argv[])
{
    //test torch
    auto device = torch::Device(torch::kCUDA);
    auto model = ConvReluBn(3,4,3);
    model->to(device);
    auto input = torch::zeros({1,3,12,12},torch::kFloat).to(device);
    auto output = model->forward(input);
    std::cout<<output.sizes()<<std::endl;

    //test opencv
    cv::Mat image = cv::imread("C:\\Users\\Administrator\\Pictures\\1.jpg");
    cv::Mat M(200, 200, CV_8UC3, cv::Scalar(0, 0, 255));
    if(!M.data)
        return 0;
    cv::imshow("fff",image);
    cv::imshow("ddd",M);
    cv::waitKey(0);
    cv::destroyAllWindows();
}

```

```
//test qt
QApplication a(argc, argv);
MainWindow w;
w.show();
return a.exec();
}
```

运行成功，并且显示200*200的红色图案。

6. 然而，仍然存在两个问题，一个是cout无输出，一个是无法调试。解决办法为：第一个在.pro文件后添加一句

```
CONFIG += console
```

第二个则相对复杂，参考[博客](#)修改。如果需要查看变量，则用博客中方法修改QT安装路径下的msvc-desktop.conf，如果不需要查看变量，能断点就行，则直接在.pro文件添加：

```
QMAKE_CXXFLAGS_RELEASE = $$QMAKE_CFLAGS_RELEASE_WITH_DEBUGINFO
QMAKE_CFLAGS_RELEASE = $$QMAKE_LFLAGS_RELEASE_WITH_DEBUGINFO
QMAKE_LFLAGS_RELEASE = $$QMAKE_LFLAGS_RELEASE_WITH_DEBUGINFO
```

至此，配置完成。

References

[OpenCV+MingW编译](#)

[QT Creator安装](#)

[win10 64bit+ qt5.12.0 + msvc2017 + libtorch\(pytorch\)+cuda10.1+cudnn7.6配置](#)

[QT creator Release模式下可调试](#)