Win11下Qt6.4.0+OpenCV4.6.0+Contrib配置编译安装开发详细步骤

一、系统环境

1.Windows11 64位系统

2.Qt6.4.0

3.CMake3.23.2(Qt系统自带)

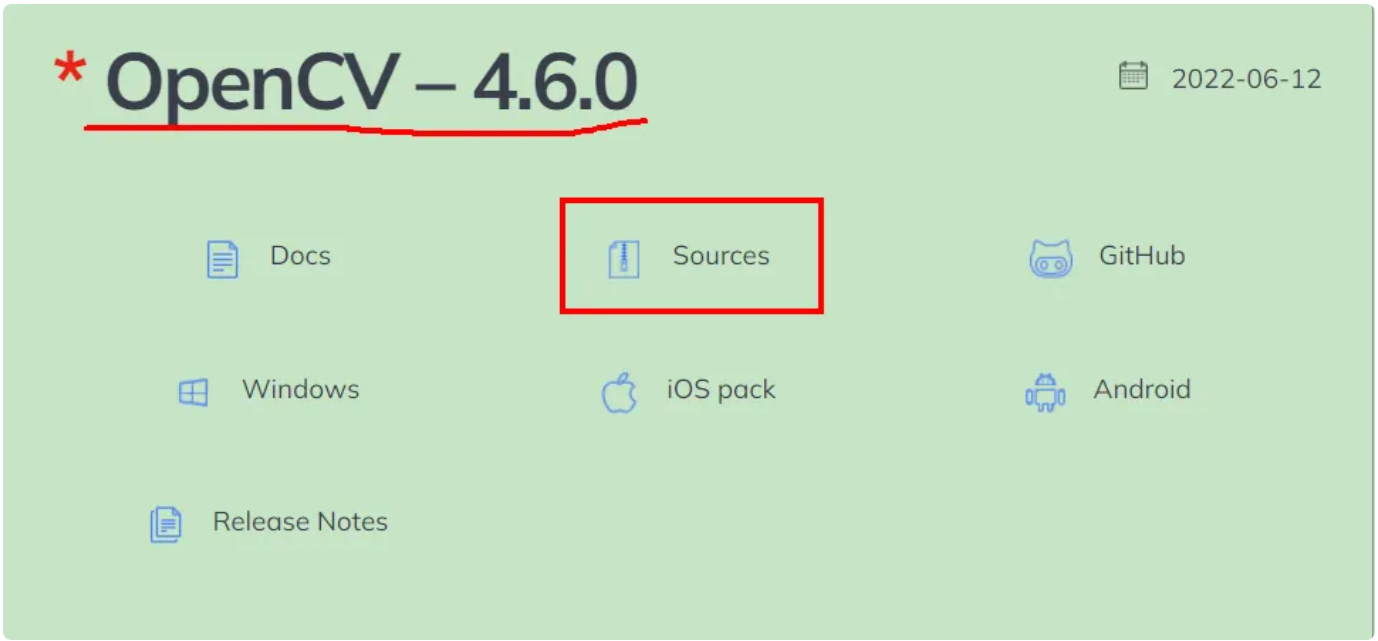
4.OpenCV4.6.0

5.Contrib

6.Python3.10.7

二、软件准备

1.下载OpenCV，网址：https://opencv.org/releases/



2.下载Contrib，网址：https://github.com/opencv

OpenCV官方已经将稳定成熟的功能都放在OpenCV包里发布，而正在发展尚未成熟的技术则统一置于Contrib扩展模块中。通常情况下，下载的OpenCV中不包含Contrib扩展库的内容，如果只是进行一般的图片、视频处理，则仅使用OpenCV就足够了。但是，OpenCV中默认不包含SIFT、SURF等先进的图像特征检测技术，另外

一些高级功能（如人脸识别等）都在Contrib扩展库中，若要充分发挥OpenCV的强大功能，则必须将其与Contrib扩展库放在一起联合编译使用。

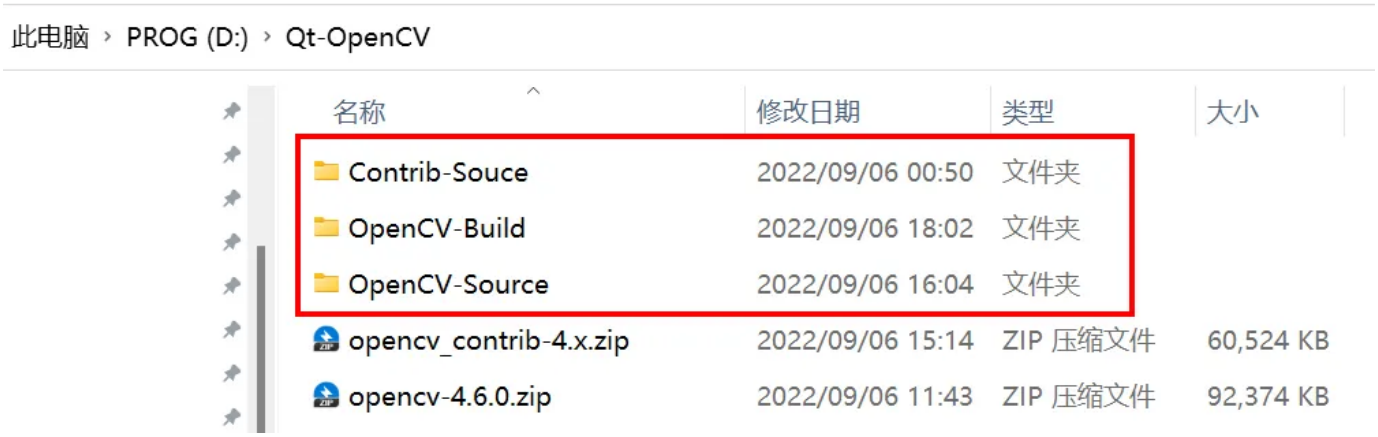
三、编译前准备

1.准备目录

（1）新建D:\Qt-OpenCV\OpenCV-Source文件夹，将下载得到的OpenCV库的opencv-4.6.0.zip压缩包解压，将得到的所有文件复制到该文件夹中。

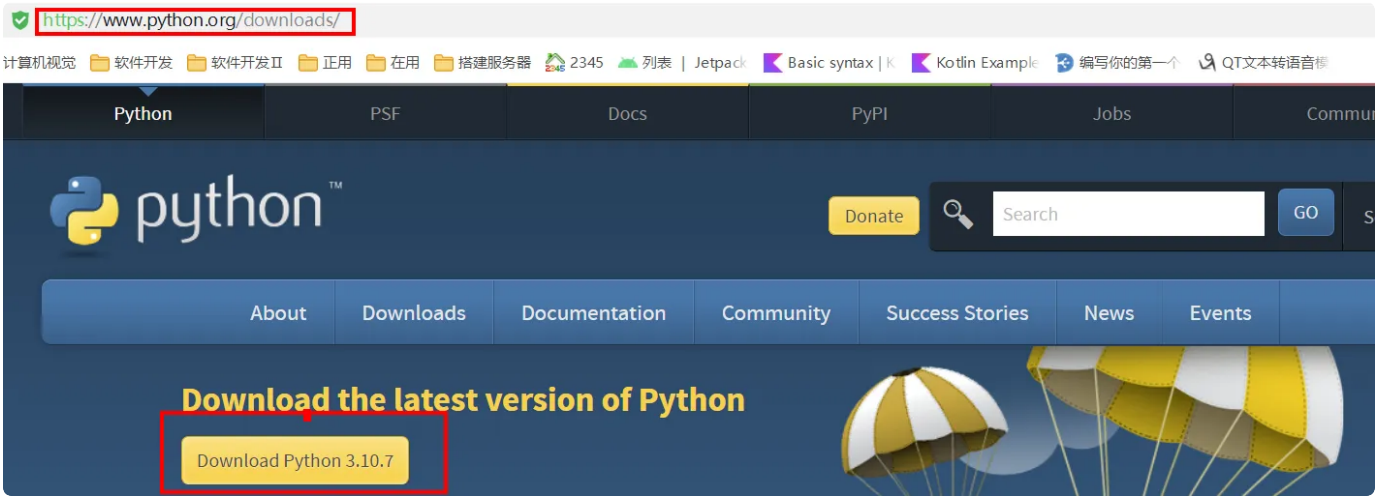
（2）新建D:\Qt-OpenCV\Contrib-Source文件夹，将下载得到的Contrib扩展库的opencv_contrib-4.xzip压缩包解压，将得到的所有文件复制到该文件夹中。

（3）新建D:\Qt-OpenCV\OpenCV-Build文件夹。



2.安装Python，官网地址：https://www.python.org/

由于OpenCV库的某些功能模块的运行环境依赖于Python平台，故编译前还要在自己的计算机系统中安装Python语言。从Python官网下载获得安装包“python-3.10.7-amd64.exe”，双击启动安装向导，采用默认配置安装即可。

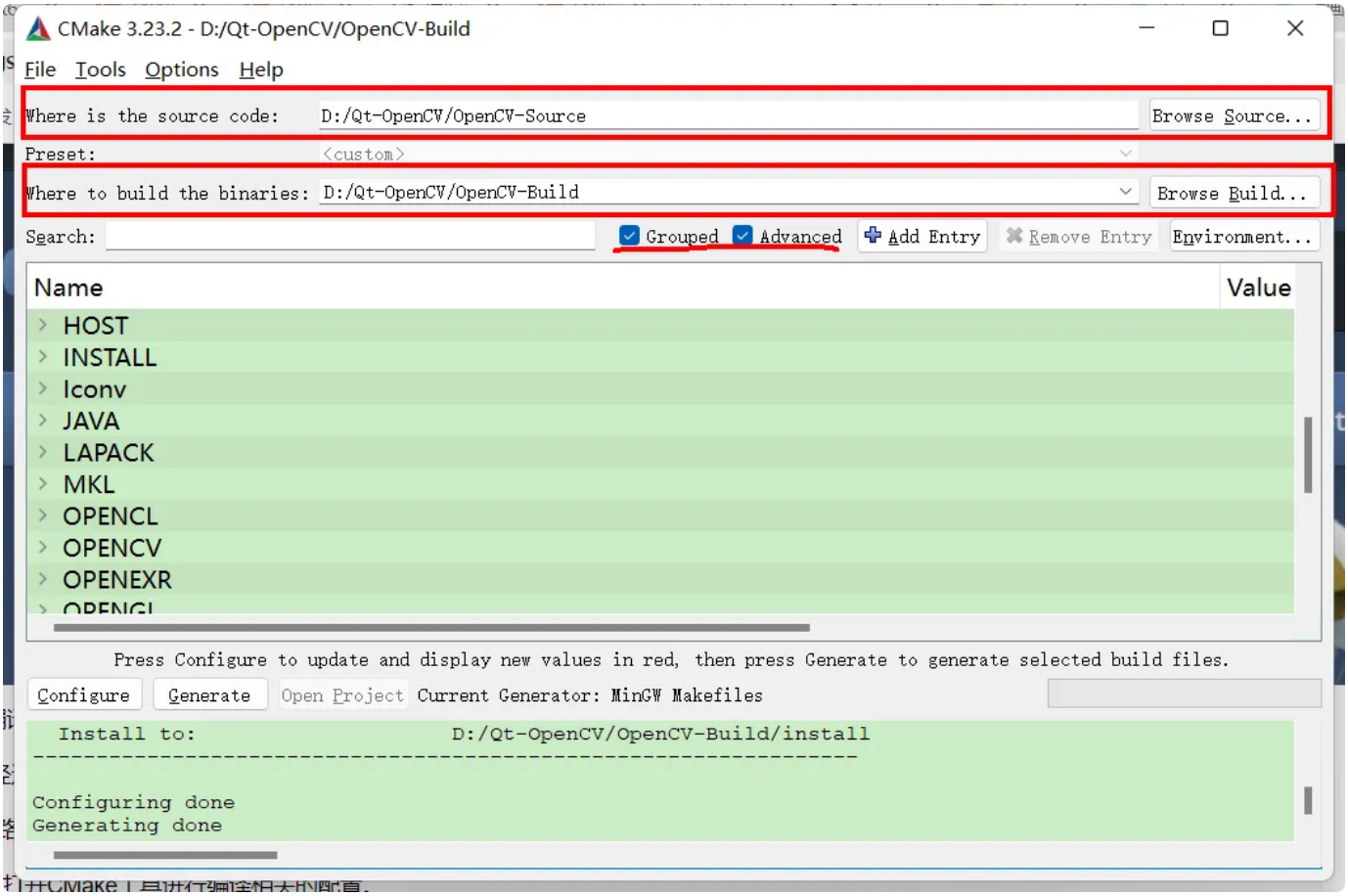


四、编译配置

经过以上各步的前期准备后，就可以正式开始编译了。

1.设置路径

首先打开CMake工具进行编译相关的配置。

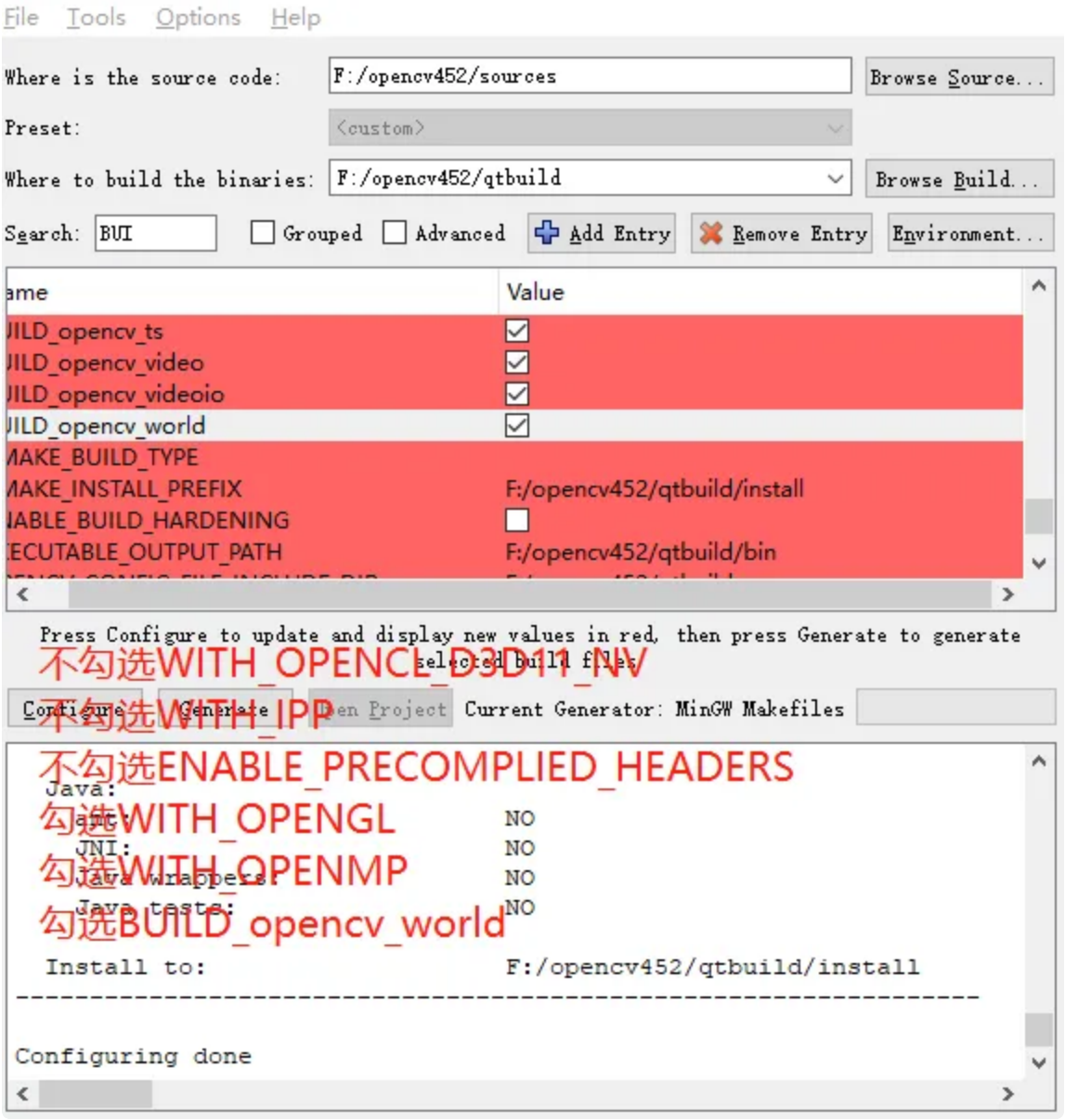


2.选择编译器

设置好路径后，单击左下角的“Configure”按钮，Specify the generator for this project:选择“MinGW Makefiles”，点选“Use default native compilers”。单击“Finish”按钮回到CMake主界面。此时，点击主界面上的“Configure”按钮，编译器配置进行中，随后，在主界面中央生成一系列红色加亮选项条的列表，同时下方信息栏输出“Configuring done”，表示编译器配置完成。

3.设置编译选项

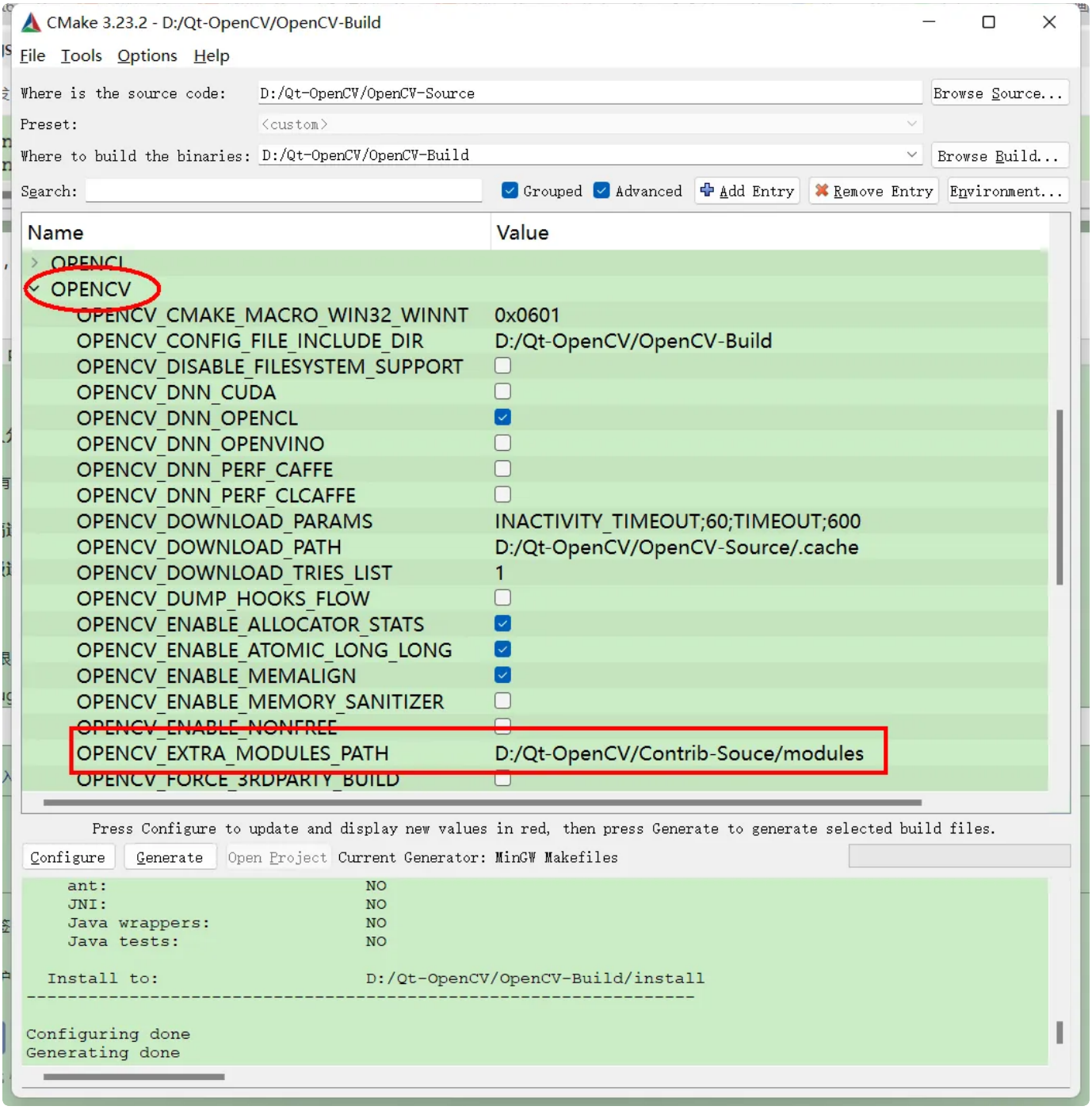
这些红色加亮的选项并非都是必须的功能，需要勾选的配置与不需要勾选的配置看红色文字，**一定要记得选中“BUILD _opencv_world”**，这将允许把所有OpenCV模块构建到一个库中（这样做的好处是在部署计算机视觉应用程序时，只需要一个DLL文件就可以了），**注意不要勾选WITH_QT！！！。**



另外，为了将Contrib扩展库与OpenCV无缝整合，还需要设置OpenCV的外接模块路径，找到名为“OPENCV_EXTRA_MODULES_PATH”的选项，设置其值为“D:/Qt-OpenCV/Contrib-Souce/modules”。

设置完成后，再次点击“Configure”按钮，界面上的红色加亮的选项全部消失，同时在下方信息栏中输出“Configuring done”；此时，点击“Generate”按钮，同时在下方信息栏中输出“Generating done”，表示编译选项全部配置完成。

提示：如果此时CMake主界面上仍然存在红色加亮的选项，则表示配置过程发生异常。解决办法是，再次单击“Configure”按钮重新进行配置，直到所有的红色加亮的选项完全消失为止。



五、开始编译

所有的设置项都完成后，就可以开始编译了。打开Windows命令行，进入到事先建好的编译生成目标目录“D:\Qt-OpenCV\OpenCV-Build”下，输入编译命令：mingw32-make -j 8，启动编译过程，在进度显示100%时，出现“Built target opencv_model_diagnostics”信息，表示编译成功，如下图所示。

```
[100%] Linking CXX executable ..\..\bin\opencv_interactive-calibration.exe
[100%] Built target opencv_interactive-calibration
[100%] Building CXX object apps/version/CMakeFiles/opencv_version.dir/opencv_version.cpp.obj
[100%] Linking CXX executable ..\..\bin\opencv_version.exe
[100%] Built target opencv_version
[100%] Building CXX object apps/version/CMakeFiles/opencv_version_win32.dir/opencv_version.cpp.obj
[100%] Linking CXX executable ..\..\bin\opencv_version_win32.exe
[100%] Built target opencv_version_win32
[100%] Building CXX object apps/model-diagnostics/CMakeFiles/opencv_model_diagnostics.dir/model_diagnostics.cpp.obj
[100%] Linking CXX executable ..\..\bin\opencv_model_diagnostics.exe
[100%] Built target opencv_model_diagnostics
```

六、安装OpenCV库

编译完成后的OpenCV库必须在安装后才能使用，在命令行中输入：mingw32-make install，安装OpenCV库，如下图。

```
Consolidate compiler generated dependencies of target opencv_interactive-calibration
[100%] Built target opencv_interactive-calibration
Consolidate compiler generated dependencies of target opencv_version
[100%] Built target opencv_version
Consolidate compiler generated dependencies of target opencv_version_win32
[100%] Built target opencv_version_win32
Consolidate compiler generated dependencies of target opencv_model_diagnostics
[100%] Built target opencv_model_diagnostics
Install the project...
-- Install configuration: "Release"
-- Installing: D:/Qt-OpenCV/OpenCV-Build/install/etc/licenses/opencl-headers-LICENSE.txt
-- Installing: D:/Qt-OpenCV/OpenCV-Build/install/etc/licenses/ade-LICENSE
```

进入“D:\Qt-OpenCV\OpenCV-Build\install”子目录，进入其中即“D:\Qt-OpenCV\OpenCV-Build\install\x64\mingw\bin”文件夹的所有文件就是编译安装好的OpenCV库文件。最终得到的OpenCV库如下图所示。

七、遇到的坑

1.设置编译选项时，若勾选“WITH_QT”，编译过程会出现如下错误信息而终止编译。所以，切记不能勾选“WITH_QT”！！！

```
[ 59%] Building CXX object modules/calib3d/CMakeFiles/opencv_perf_calib3d.dir/perf/perf_pnp.cpp.obj
[ 59%] Building CXX object modules/calib3d/CMakeFiles/opencv_perf_calib3d.dir/perf/perf_stereosgbm.cpp.obj
[ 60%] Building CXX object modules/calib3d/CMakeFiles/opencv_perf_calib3d.dir/perf/perf_undistort.cpp.obj
[ 60%] Linking CXX executable ..\..\bin\opencv_perf_calib3d.exe
[ 60%] Built target opencv_perf_calib3d
[ 60%] Automatic MOC for target opencv_cvv
[ 60%] Built target opencv_cvv_autogen
Scanning dependencies of target opencv_cvv
[ 60%] Building CXX object modules/cvv/CMakeFiles/opencv_cvv.dir/opencv_cvv_autogen/mocs_compilation.cpp.obj
In file included from D:\Qt-OpenCV\OpenCV-Build\modules\cvv\opencv_cvv_autogen\MXUWEOXILK/moc_call_tab.cpp:10,
                 from D:\Qt-OpenCV\OpenCV-Build\modules\cvv\opencv_cvv_autogen\mocs_compilation.cpp:2:
D:/Qt-OpenCV/Contrib-Souce/modules/cvv/src/gui/call_tab.hpp:4:10: fatal error: QString: No such file or directory
    4 | #include <QString>

compilation terminated.
mingw32-make[2]: *** [modules\cvv\CMakeFiles\opencv_cvv.dir\build.make:76: modules/cvv/CMakeFiles/opencv_cvv.dir/opencv_cvv_autogen/mocs_compilation.cpp.obj] Error 1
mingw32-make[1]: *** [CMakeFiles\Makefile2:5628: modules/cvv/CMakeFiles/opencv_cvv.dir/all] Error 2
mingw32-make: *** [Makefile:165: all] Error 2
```

2.cmake中Configure时 ffmpeg.dll 下载一直失败（这会导致编译后的OpenCV无法用VideoCapture打开视频）等问题的解决方法可参考如下网址：

https://wangjichuan.blog.csdn.net/article/details/110677323?spm=1001.2101.3001.6661.1&depth_1-utm_relevant_index=1

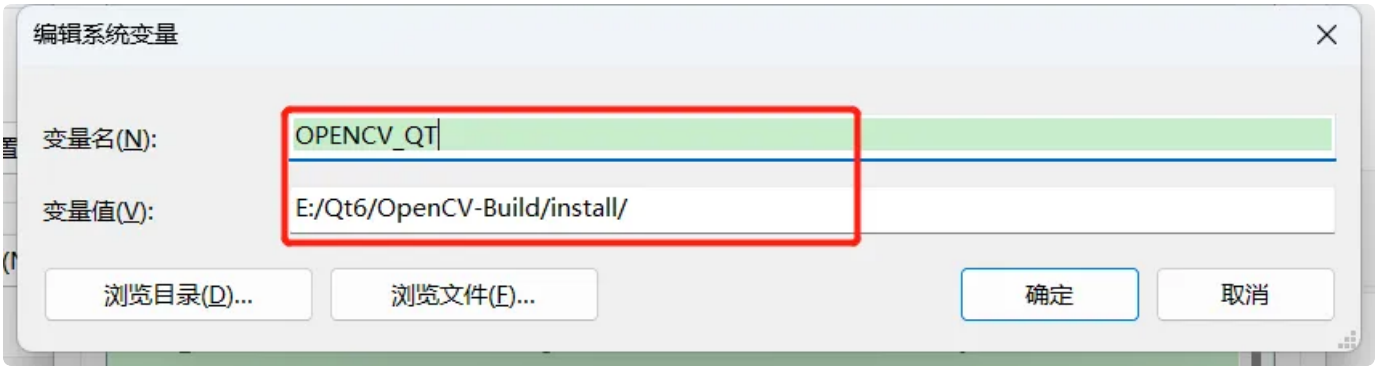
3.编辑C:\Windows\System32\drivers\etc目录下的hosts文件，添加如下内容：

2606:50c0:8002::154 raw.githubusercontent.com

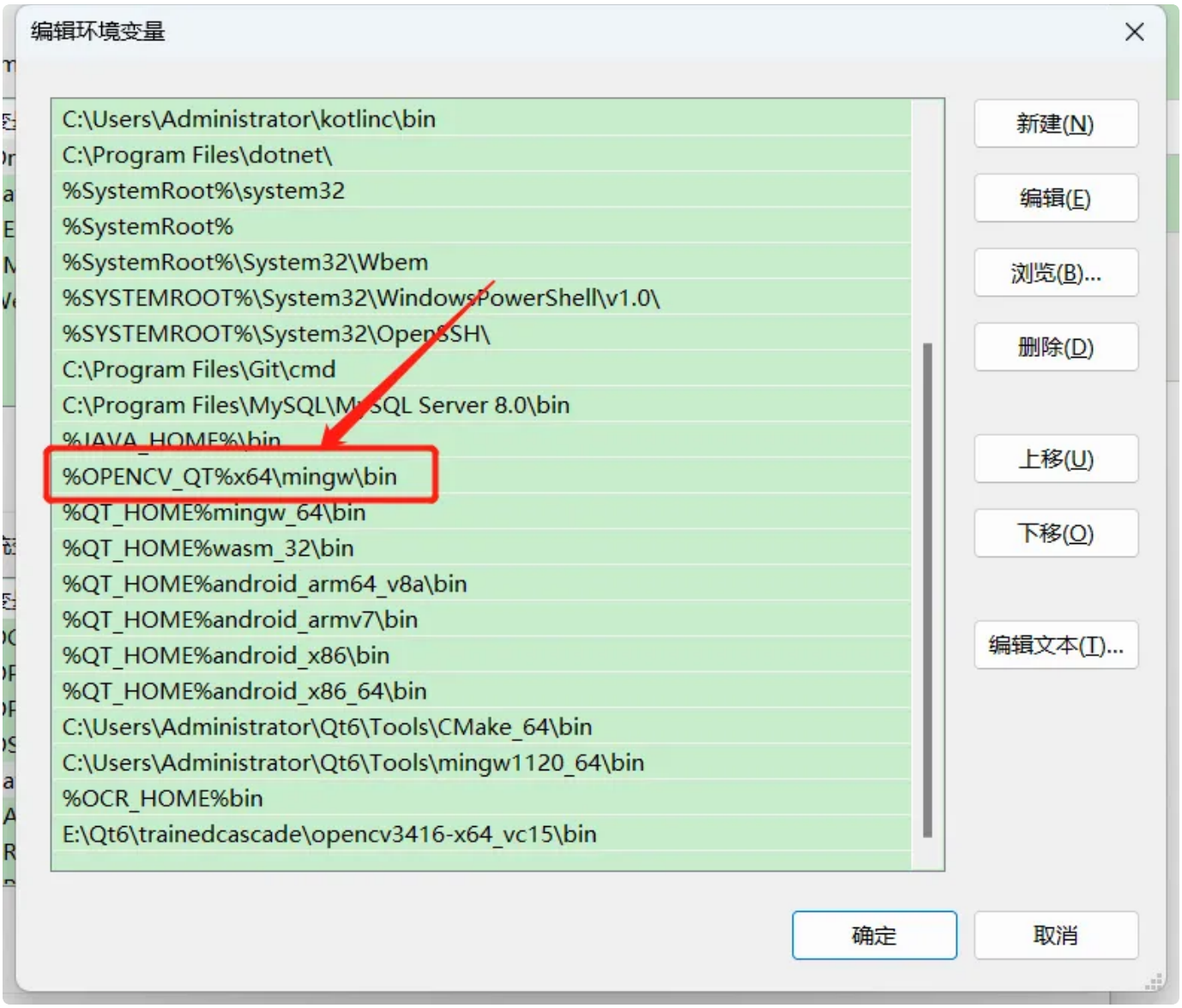
raw.githubusercontent.com 对应的具体内容可在 https://tool.chinaz.com/speedtest/gist.github.com测试后编写。

八、设置系统环境变量

1.创建系统环境变量“OPENCV_QT”。



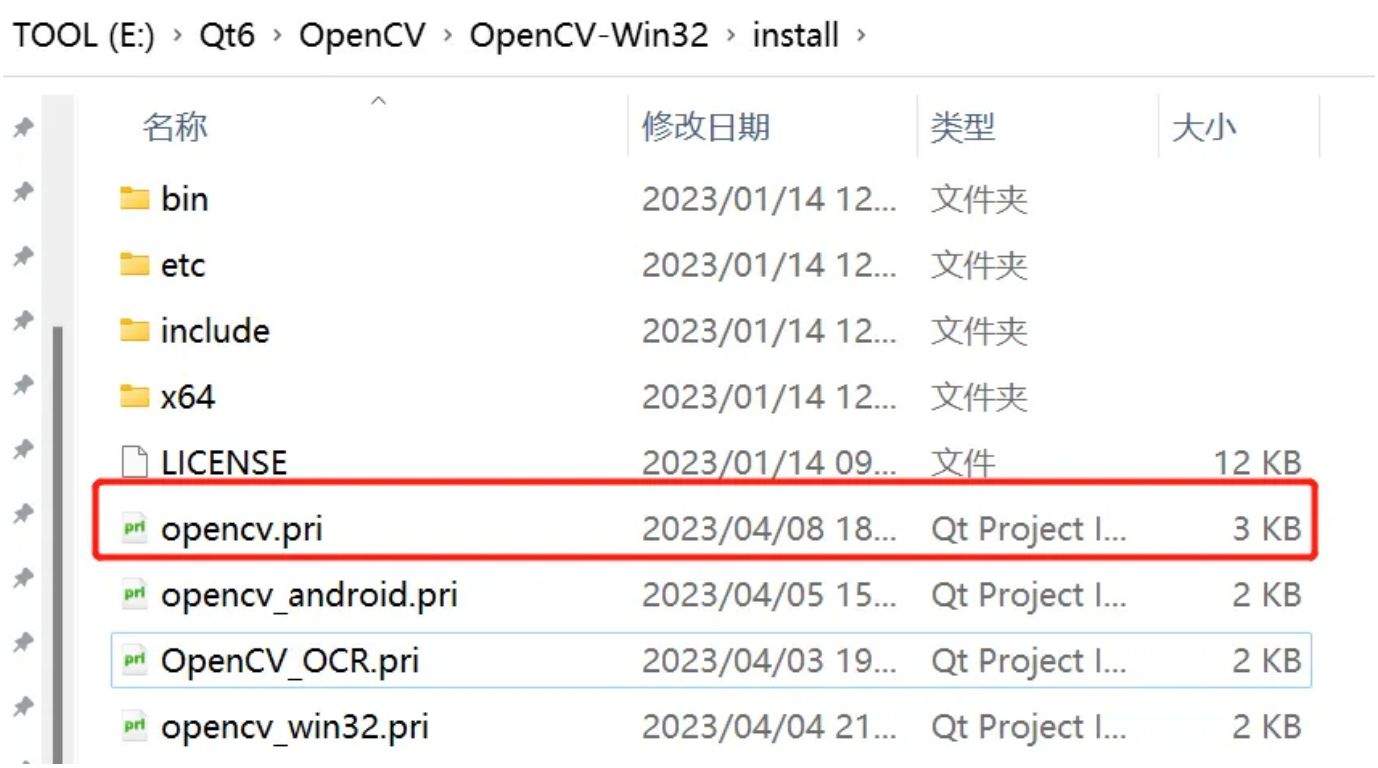
2.将新生成的“D:\Qt-OpenCV\OpenCV-Build\install\x64\mingw\bin”添加到Path变量。Qt、CMake、OpenCV环境变量设置如下图。



3.在“D:\Qt-OpenCV\OpenCV-Build\install”目录下创建“opencv.pri”文件，内容如下。

```
# 用来描述构建项目时应该搜索的 include 目录。
INCLUDEPATH += $$(OPENCV_HOME)include \
$$(OPENCV_HOME)include\opencv \
$$(OPENCV_HOME)include\opencv2
# 用来描述项目引用的库文件列表。
LIBS += $$(OPENCV_HOME)x64/mingw/lib/libopencv_*.a
```

4.“D:\Qt-OpenCV\OpenCV-Build\install”目录，如下图。



八、测试程序

1.在项目.pro文件中，引入.pri文件（复制粘贴如下内容即可）。

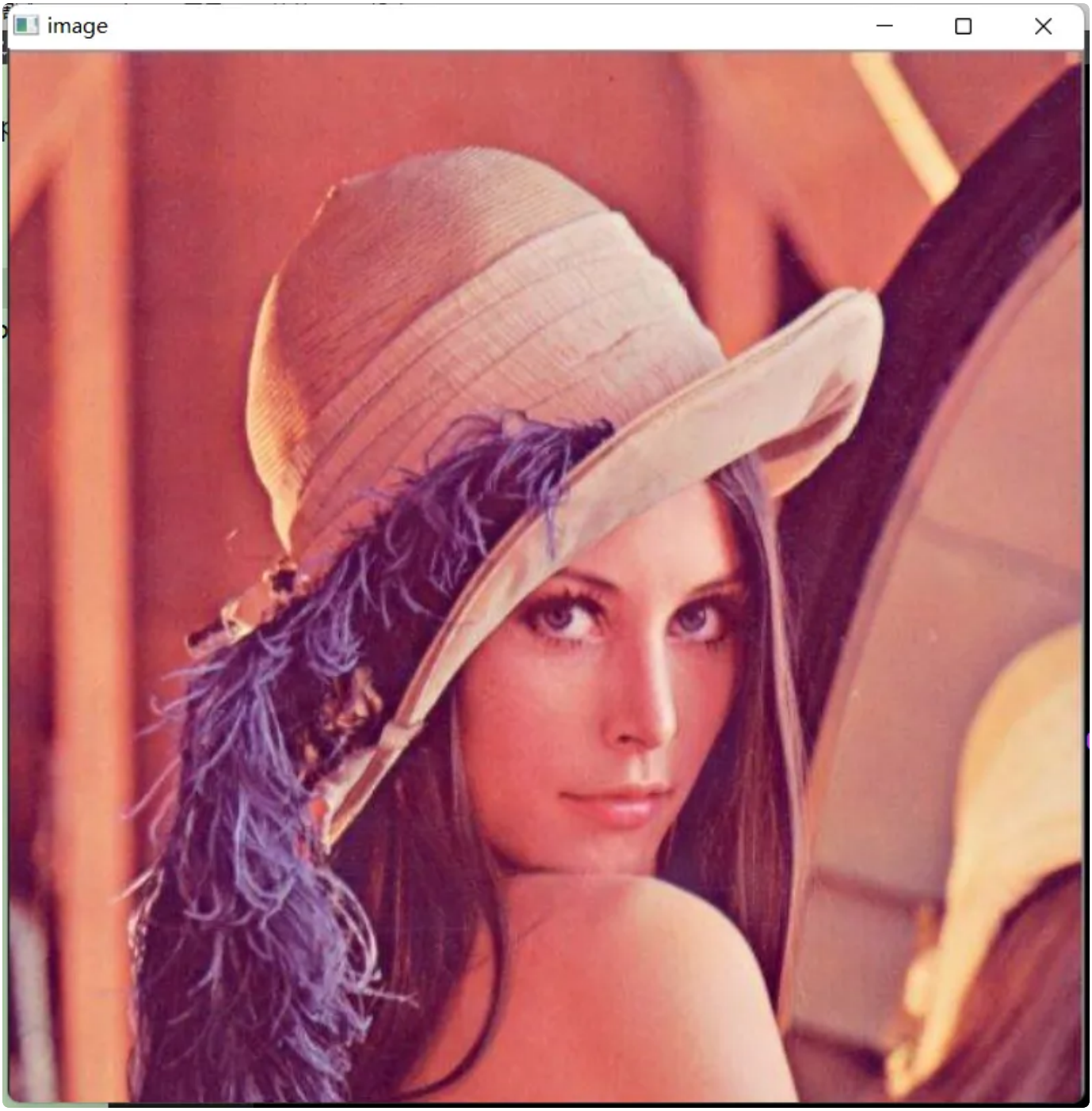
```
# Qt + OpenCV 应用程序
OPENCV = $$(OPENCV_QT)
if(!isEmpty(OPENCV)) {
#include($$(OPENCV_QT)opencv.pri)
include($$(OPENCV_QT)opencv.pri)
} else {
error(OPENCV_QT does not exist!!!)
}
```

```
5  QT += core gui multimedia
6  greaterThan(QT_MAJOR_VERSION, 4): QT += widgets
7
8  # Qt + OpenCV 应用程序
9  OPENCV = $$(OPENCV_QT)
10 if(!isEmpty(OPENCV)) {
11     include($$(OPENCV_QT)opencv.pri)
12 } else {
13     error(OPENCV_QT does not exist!!!)
14 }
15
```

2.main中代码：

```cpp
#include <QApplication>
#include <core.hpp>
#include <face.hpp>
#include <opencv.hpp> //OpenCV文件包含
#include <vector>     //包含向量类动态数组功能

using namespace cv;   // OpenCV命名空间
using namespace std;  //使用vector必须
using namespace cv::face;

int main(int argc, char *argv[])
{
    Q_UNUSED(argc);
    Q_UNUSED(argv);

    Mat img = imread("D:/Datum/Qt/Qt-Res/pic/lena.png", 1);
    namedWindow("image");
    imshow("image", img);
    waitKey(0);

    return 0;
}
```

3.运行程序，显示图片：



标签: Qt