

# Qt、MinGW、CMake 編譯 OpenCV 4.5.4（包含 opencv\_contrib）詳細過程目錄

Qt、MinGW、CMake 編譯 OpenCV 4.5.4（包含 opencv\_contrib）詳細過程

- （1）工具下載準備
- （2）CMake 編譯前組態
- （3）編譯和安裝
- （4）Qt 使用 Opencv4.5.4 測試
- （5）注意事項

## （1）工具下載準備

CMake 下載：<https://cmake.org/download/>，我使用的版本是 3.22.3

OpenCV 原始碼下載：<https://github.com/opencv/opencv/tags>

opencv\_contrib 原始碼下載：[https://github.com/opencv/opencv\\_contrib/tags](https://github.com/opencv/opencv_contrib/tags)

Qt 5.14.2，下載地址：<https://download.qt.io/archive/qt/5.14/5.14.2/>

MinGW：使用的 Qt5.14.2 自帶的 MinGW（gcc 版本為 7.3.0）

注意：編譯的 opencv\_contrib 要跟 OpenCV 的版本對應！！

問：為什麼選用 OpenCV 4.5.4 這個版本？

答：截止 2022 年 3 月 17 日，opencv 的最新版本為 4.5.5，我最初編譯使用的就是 opencv4.5.5，但是該版本的 protobuf 版本為 3.19，使用的 Qt5.14.2 自帶的 MinGW 編譯回報錯誤編譯不過去，具體錯誤為：

```
E:/opencv/opencv-4.5.5/3rdparty/protobuf/src/google/protobuf/stubs/mutex.h:124:29: error:
temporary of non-literal type 'google::protobuf::internal::CallOnceInitializedMutex<std::mutex>'
in a constant expression
    constexpr WrappedMutex() {}
```

這是由於 protobuf 這個版本太新了，Qt5.14.2 中 MinGW 的 gcc 版本還不支援這種 C++ 新語法，所以我再去查看 OpenCV 4.5.4 版本中 protobuf 版本發現是 3.5.1，這個版本舊沒有用到 C++ 語法的新特性，所以編譯沒有出現問題。

在一個原因是編譯 OpenCV 4.5.5 勾選 WITH\_OPENGL 後，編譯也過不去，還沒有找到問題所在，

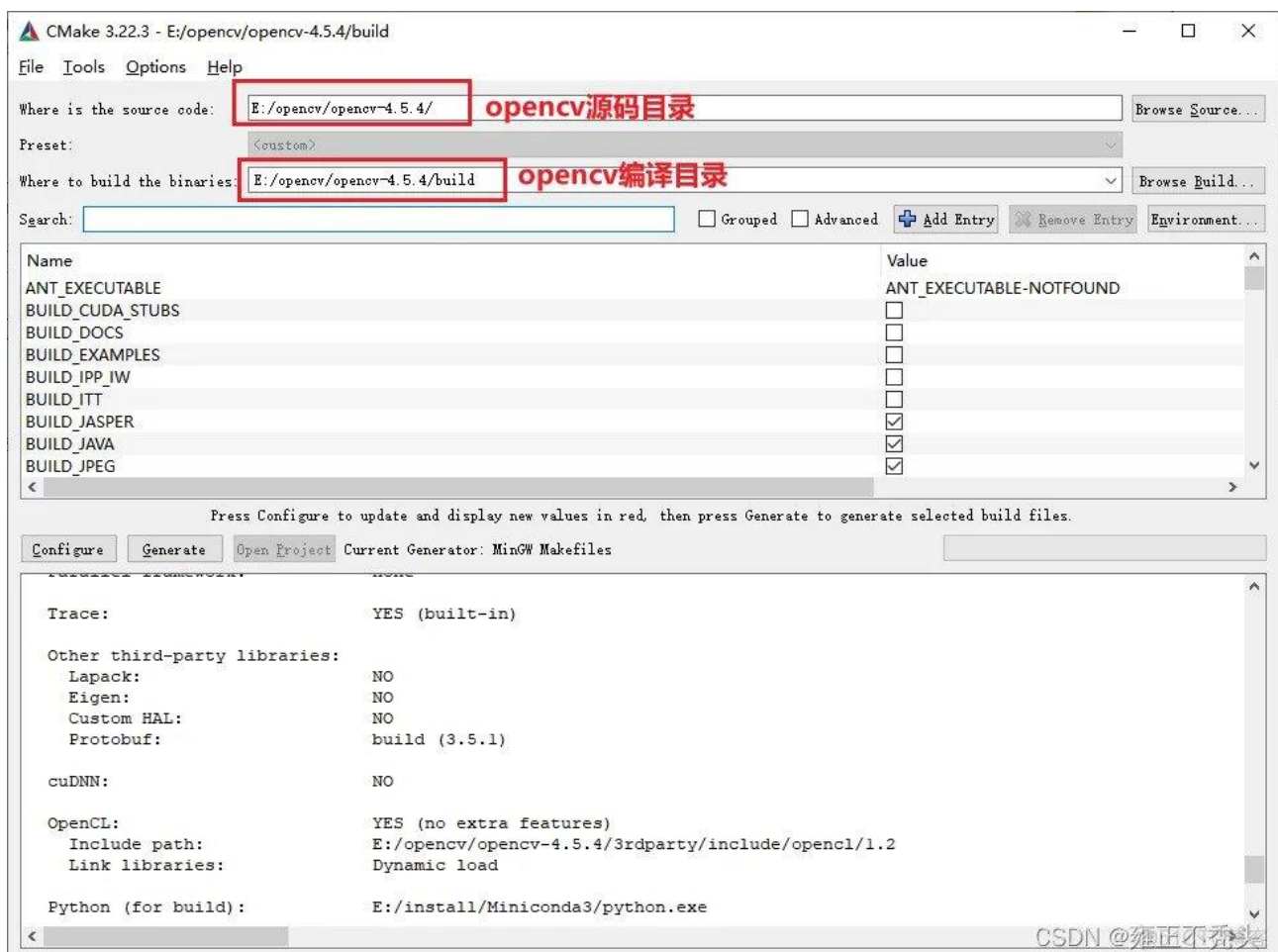
具體錯誤為：

```
[ 28%] Linking CXX shared library ..\..\bin\libopencv_highgui455.dll
CMakeFiles\opencv_highgui.dir/objects.a(window_QT.cpp.obj):window_QT.cpp:(.text$_ZN14OpenGL
Viewport12initializeGLEv+0x14): undefined reference to `_imp__glHint@8'
CMakeFiles\opencv_highgui.dir/objects.a(window_QT.cpp.obj):window_QT.cpp:(.text$_ZN14OpenGL
Viewport8resizeGLEv+0x24): undefined reference to `_imp__glViewport@16'
CMakeFiles\opencv_highgui.dir/objects.a(window_QT.cpp.obj):window_QT.cpp:(.text$_ZN14OpenGL
Viewport7paintGLEv+0xf): undefined reference to `_imp__glClear@4'
```

綜上原因，顧選定為 OpenCV 4.5.4 版本，只要使用的工具版本跟著我這步驟來，必定能編譯通過！

## (2) CMake 編譯前組態

打開 CMake，點選 Browser Source... 選擇 opencv 原始碼目錄，然後在填入 opencv 編譯組態的目錄：



然後選擇編譯 makefile 的類型問 MinGW，手動指定本地編譯工具鏈，點後點 Next：



Specify the generator for this project

MinGW Makefiles

☐ Use default native compilers

☒ Specify native compilers

☐ Specify toolchain file for cross-compiling

☐ Specify options for cross-compiling

填入 C 編譯器的目錄：E:/install/Qt/Qt5.14.2/Tools/mingw730\_32/bin/gcc.exe

填入 C++編譯器的目錄：E:/install/Qt/Qt5.14.2/Tools/mingw730\_32/bin/g++.exe

然後點 Finish 即可。



Compilers

C: E:/install/Qt/Qt5.14.2/Tools/mingw730\_32/bin/gcc.exe ... C++: E:/install/Qt/Qt5.14.2/Tools/mingw730\_32/bin/g++.exe ...

Fortran: ...

點 Finish 後會進入主介面進行組態，組態完成後必定會出現一堆紅，先不用管。

首先新增 opencv\_contrib 擴展的編譯，搜尋找到 **OPENCV\_EXTRA\_MODULES\_PATH**，填入 opencv 擴展模組的目錄：E:/opencv/opencv\_contrib-4.5.4/modules

Where is the source code: E:/opencv/opencv-4.5.4/ Browse Source...

Preset: <custom>

Where to build the binaries: E:/opencv/opencv-4.5.4/build Browse Build...

Search: EXTRA ☐ Grouped ☐ Advanced   Environment...

| Name                      | Value                                  |
|---------------------------|--|
| OPENCV_EXTRA_MODULES_PATH | E:/opencv/opencv_contrib-4.5.4/modules |

CSDN @雅正不亮爽

搜尋 WITH\_QT，並勾選，同樣的也勾選 WITH\_OPENGL：

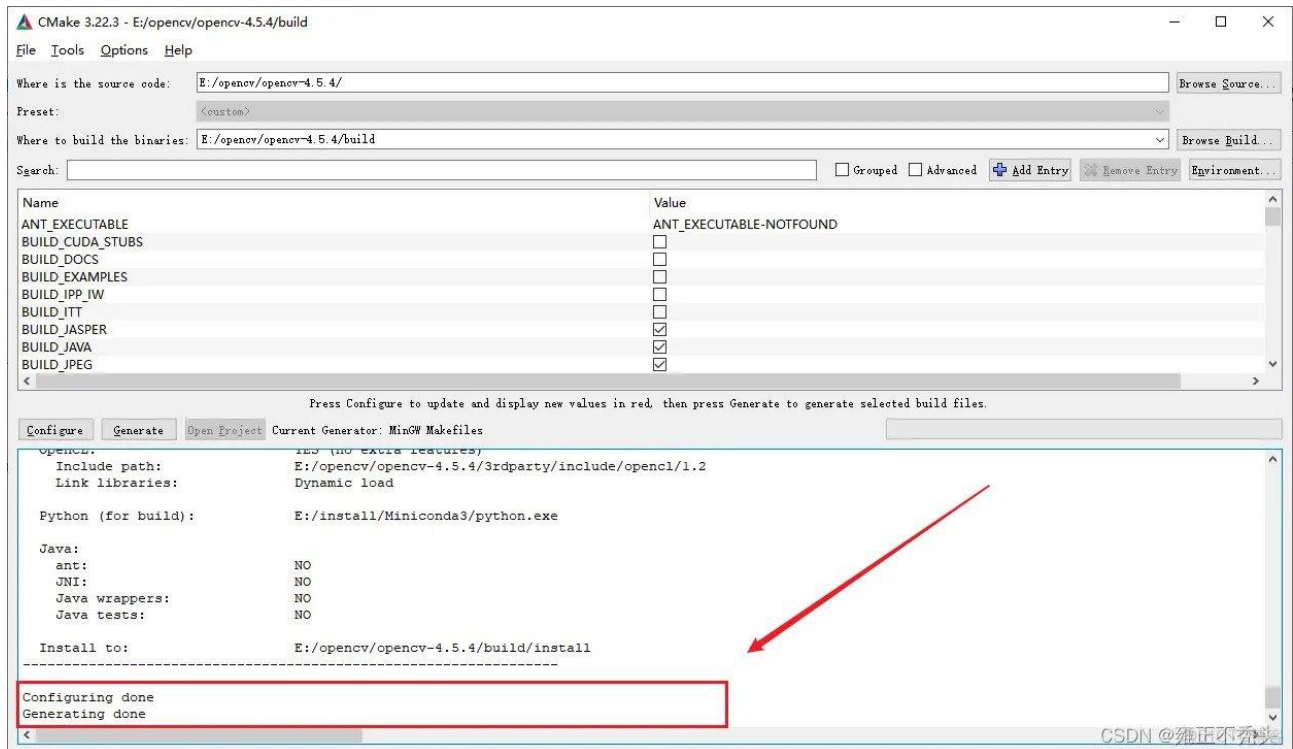
再點選 Configure，找到 QT\_QMAKE\_EXECUTABLE，填入 Qt 的 qmake 目錄：

E:/install/Qt/Qt5.14.2/5.14.2/mingw73\_32/lib/cmake/Qt5

找到 Qt5\_DIR，填入 Qt5 的目錄：E:/install/Qt/Qt5.14.2/5.14.2/mingw73\_32/bin/qmake.exe

以上就是所有組態步驟。

再點選 **Configure** 即可，有紅色顯示沒關係，多點幾次 **Configure** 就沒了，沒紅色之後，再點選 **Generate** 生成工程即可，如下圖：



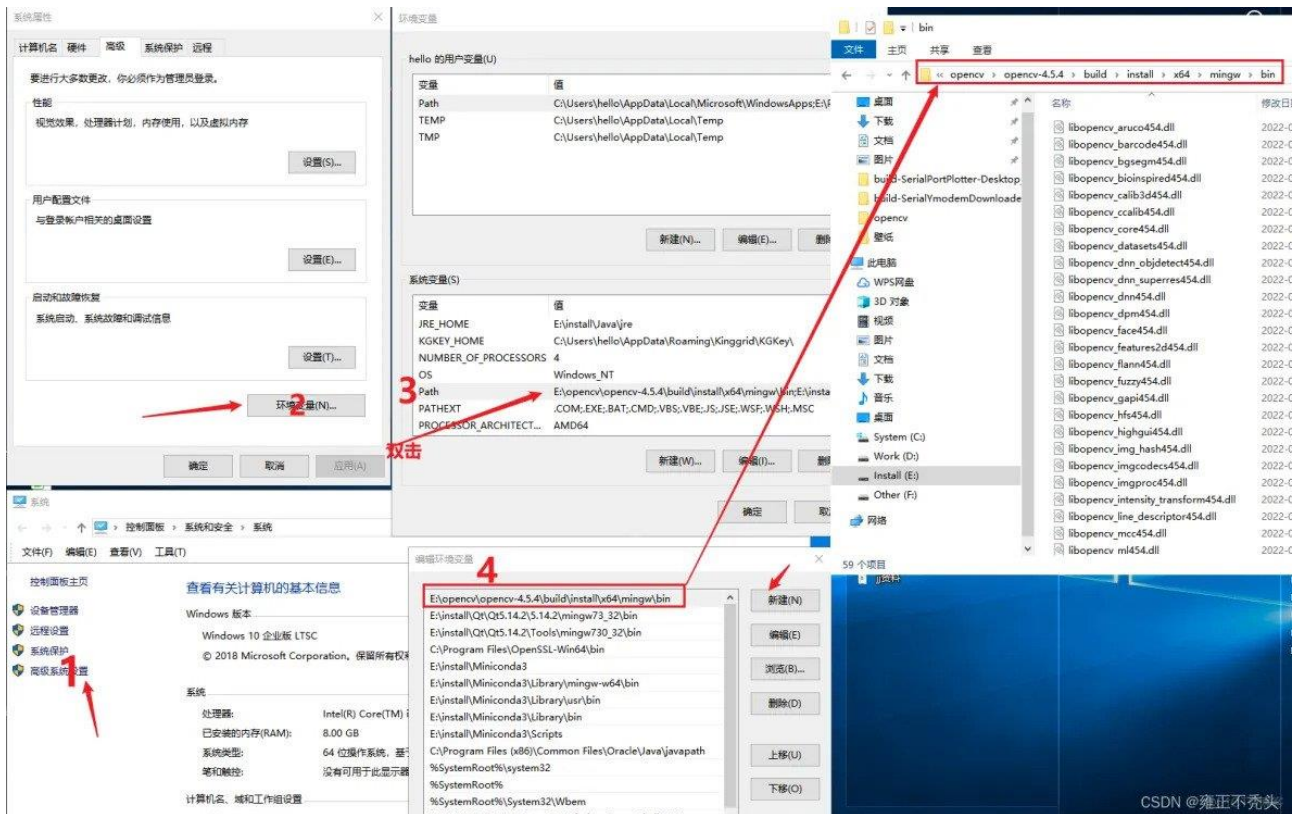
### （3）編譯和安裝

進入目錄 `E:/opencv/opencv-4.5.4/build`，執行命令：`mingw32-make -j 4` 進行編譯，`-j 4` 的意思是四核一塊編譯，這樣能夠加快編譯速度，我在編譯過程中一個問題也沒出現。

編譯完成後，執行命令 `mingw32-make install`，此時就已經默認安裝到了 `E:/opencv/opencv-4.5.4/build/install` 目錄下

### （4）Qt 使用 Opencv4.5.4 測試

在測試前，要先把剛編譯好的 `opencv 4.5.4` 庫加入系統環境變數，最好重啟一下電腦，否則 Qt 由於找不到連結庫，程式執行不起來。



新建 Qt 工程 opencv-test，在.pro 檔案中加入 opencv 的標頭檔和庫檔案包含：

```
INCLUDEPATH += E:\opencv\opencv-4.5.4\build\install\include
```

```
LIBS += E:\opencv\opencv-4.5.4\build\install\x64\mingw\lib\libopencv_*.a
```

編寫測試程序：

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <opencv2/opencv.hpp>
```

using namespace cv; // 引入 opencv 的命名空間

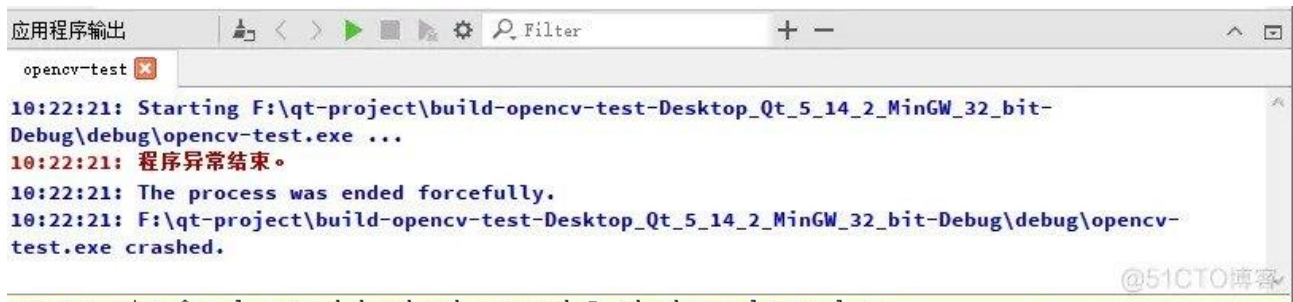
```
MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
```

```
Mat image(cv::Size(300, 300), CV_8U, Scalar(255, 0, 0)); // 300X300 大小的白色圖片
```

```
    imshow("opencv test", image);                // 顯示圖片
}

MainWindow::~MainWindow()
{
    delete ui;
}
```

編譯運行，程序啟動不起來：



這就是由於 Qt 找不到 opencv 的動態連結庫導致的，由於我沒有重啟電腦，所以這裡我直接將目錄 E:\opencv\opencv-4.5.4\build\install\x64\mingw\bin 下的.dll 檔案直接複製到了 opencv-test.exe 目錄下，然後在運行 Qt 程序，正常運行了！

## （5）注意事項

在使用 CMake 點選 Configure 後，會下載一些必要的檔案，而且下載的過程很慢，務必保持網路通暢！

有時候一些檔案一直下載不下來，那麼我的解決辦法是，Configure 完成之後，會在 E:\opencv\opencv-4.5.4\build 目錄下生成一個 CMakeDownloadLog.txt 檔案，該檔案中記錄了所需下載檔案的網址，你可以手動下載然後放進相應的目錄中！

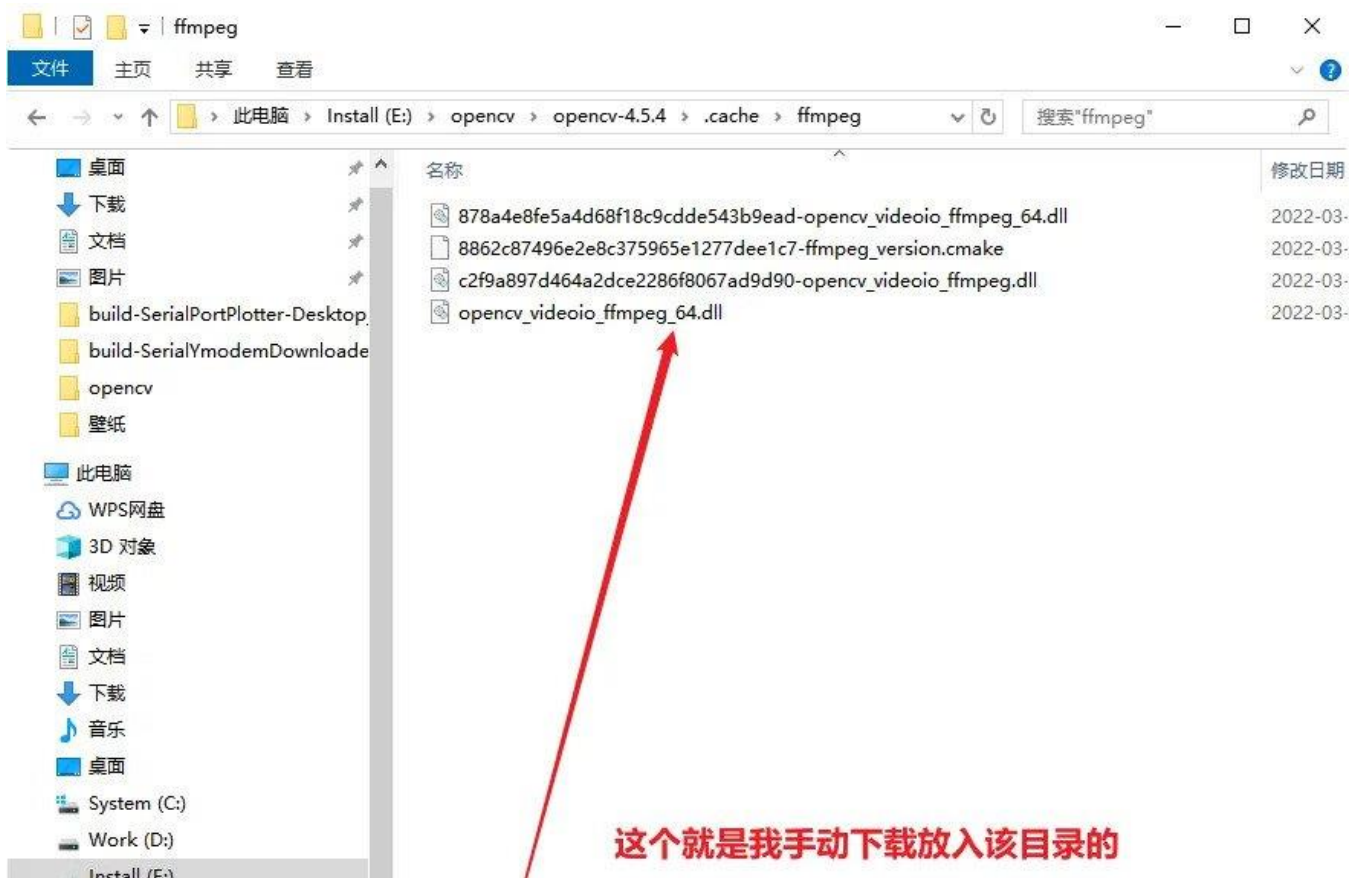
以 opencv\_videoio\_ffmpeg\_64.dll 這個檔案為例，操作步驟如下：



```
1 #use_cache "E:/opencv/opencv-4.5.4/.cache"
2 #match_hash_in_cmake_cache "OCV_DOWNLOAD_ADE_HASH_3rdparty_ade_v0.1_if_zip"
3 #match_hash_in_cmake_cache "OCV_DOWNLOAD_FFMPEG_HASH_3rdparty_ffmpeg_opencv_videoio_ffmpeg_dll"
4 #do_gnpy "opencv_videoio_ffmpeg_64.dll" "878a4e8fe5a4d68f18c9cdde543b9ead-opencv_videoio_ffmpeg_64.dll" "https://raw.githubusercontent.com/opencv/opencv_3rdparty/40b466d1aa374205fde1373496e15d92ecd5313/ffmpeg/opencv_videoio_ffmpeg_64.dll" "E:/opencv/opencv-4.5.4/build/downloads/ffmpeg_64.dll"
5 #missing "E:/opencv/opencv-4.5.4/build/3rdparty/ffmpeg/opencv_videoio_ffmpeg_64.dll"
6 #check_md5 "E:/opencv/opencv-4.5.4/.cache/ffmpeg/878a4e8fe5a4d68f18c9cdde543b9ead-opencv_videoio_ffmpeg_64.dll" "d9ca7b6113ed9c5e72479c2d9580b387"
7 #ismatch_md5 "E:/opencv/opencv-4.5.4/.cache/ffmpeg/878a4e8fe5a4d68f18c9cdde543b9ead-opencv_videoio_ffmpeg_64.dll"
8 #delete "E:/opencv/opencv-4.5.4/.cache/ffmpeg/878a4e8fe5a4d68f18c9cdde543b9ead-opencv_videoio_ffmpeg_64.dll" "https://raw.githubusercontent.com/opencv/opencv_3rdparty/40b466d1aa374205fde1373496e15d92ecd5313/ffmpeg/opencv_videoio_ffmpeg_64.dll"
9 #cmake_download "E:/opencv/opencv-4.5.4/.cache/ffmpeg/878a4e8fe5a4d68f18c9cdde543b9ead-opencv_videoio_ffmpeg_64.dll" "https://raw.githubusercontent.com/opencv/opencv_3rdparty/40b466d1aa374205fde1373496e15d92ecd5313/ffmpeg/opencv_videoio_ffmpeg_64.dll"
10 #try 1
11 #match_hash_in_cmake_cache "OCV_DOWNLOAD_FFMPEG_HASH_3rdparty_ffmpeg_ffmpeg_version_cmake"
12 #match_hash_in_cmake_cache "OCV_DOWNLOAD_WECHAT_QRCODE_HASH_downloads_wechat_qrcode_detect_caffemodel"
13 #match_hash_in_cmake_cache "OCV_DOWNLOAD_WECHAT_QRCODE_HASH_downloads_wechat_qrcode_detect_prototxt"
14 #do_gnpy "sr.caffemodel" "cbfd60361a73beb8c583eea7e5e6664-sr.caffemodel" "https://raw.githubusercontent.com/WeChatCV/opencv_3rdparty/a8b69cc738421293254aec5dd38bd523503252/sr.caffemodel" "E:/opencv/opencv-4.5.4/build/downloads/wechat_qrcode"
15 #missing "E:/opencv/opencv-4.5.4/build/downloads/wechat_qrcode/sr.caffemodel"
16 #check_md5 "E:/opencv/opencv-4.5.4/.cache/wechat_qrcode/cbfd60361a73beb8c583eea7e5e6664-sr.caffemodel" "d41d8cd98f00b204e9800998ecf8427e"
17 #ismatch_md5 "E:/opencv/opencv-4.5.4/.cache/wechat_qrcode/cbfd60361a73beb8c583eea7e5e6664-sr.caffemodel" "d41d8cd98f00b204e9800998ecf8427e"
18 #delete "E:/opencv/opencv-4.5.4/.cache/wechat_qrcode/cbfd60361a73beb8c583eea7e5e6664-sr.caffemodel"
19 #cmake_download "E:/opencv/opencv-4.5.4/.cache/wechat_qrcode/cbfd60361a73beb8c583eea7e5e6664-sr.caffemodel" "https://raw.githubusercontent.com/WeChatCV/opencv_3rdparty/a8b69cc738421293254aec5dd38bd523503252/sr.caffemodel"
20 #try 1
21 # timeout on name lookup is not supported
22 # Trying 185.199.109.133:443...
23 # Connected to raw.githubusercontent.com (185.199.109.133) port 443 (#0)
24 # channel: disabled automatic use of client certificate
25 # schannel: ALPN, offering h2
26 # schannel: ALPN, offering http/1.1
27 # schannel: ALPN, server accepted to use h2
28 # Using HTTP2, server supports multiplexing
29 # Connection state changed (HTTP/2 confirmed)
30 # Copying HTTP/2 data in stream buffer to connection buffer after upgrade: len=0
31 # Using Stream ID: 1 (easy handle 0x1794e359a0)
32 # GET /WeChatCV/opencv_3rdparty/a8b69cc738421293254aec5dd38bd523503252/sr.caffemodel HTTP/2
33
```

下载地址

需要放到该目录下



这个就是我手动下载放入该目录的

ends...

[https://blog.51cto.com/u\\_15950551/6031899](https://blog.51cto.com/u_15950551/6031899)