**C++语言编程技巧**

目录

[1. Opencv库 1](#_Toc157154270)

[1.1. Visual Studio配置Opencv C++ 1](#_Toc157154271)

[1.2. 函数Mat坐标参数顺序 4](#_Toc157154272)

[1.3. 从文件夹中批量读取文件 5](#_Toc157154273)

[2. C++小技巧 5](#_Toc157154274)

[2.1. VS编译及调试相关 5](#_Toc157154275)

[2.2. 获取代码运行时间 5](#_Toc157154276)

[2.3. 文件绝对路径与相对路径 6](#_Toc157154277)

[2.4. 数值变量转换为字符串 6](#_Toc157154278)

[2.5. 字符串转换为数值变量 6](#_Toc157154279)

[2.6. 循环结构的临时变量 7](#_Toc157154280)

[2.7. 编程规范相关 7](#_Toc157154281)

[3. FAQ 7](#_Toc157154282)

# Opencv库

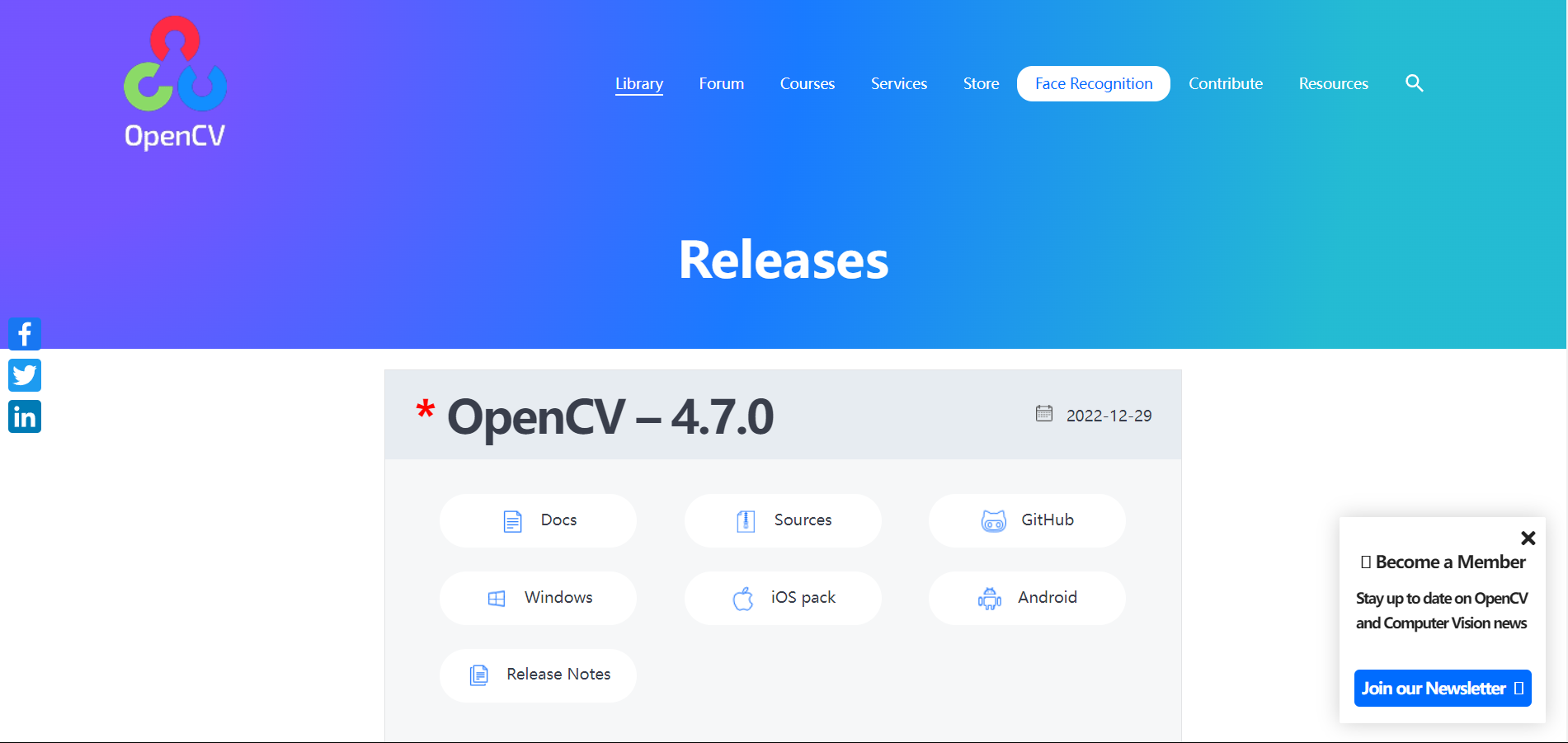
Opencv全称为“开源计算机视觉库”，是由Intel开发的功能强大的跨平台即时视觉库，支持语言包括C++和Python，内含数字图像处理、机器视觉的常用函数以及用于储存图像的Mat类容器。

## Visual Studio配置Opencv C++

·基本流程：Opencv下载安装、配置系统变量、配置VS项目属性、代码引用头文件

1. Opencv下载及安装

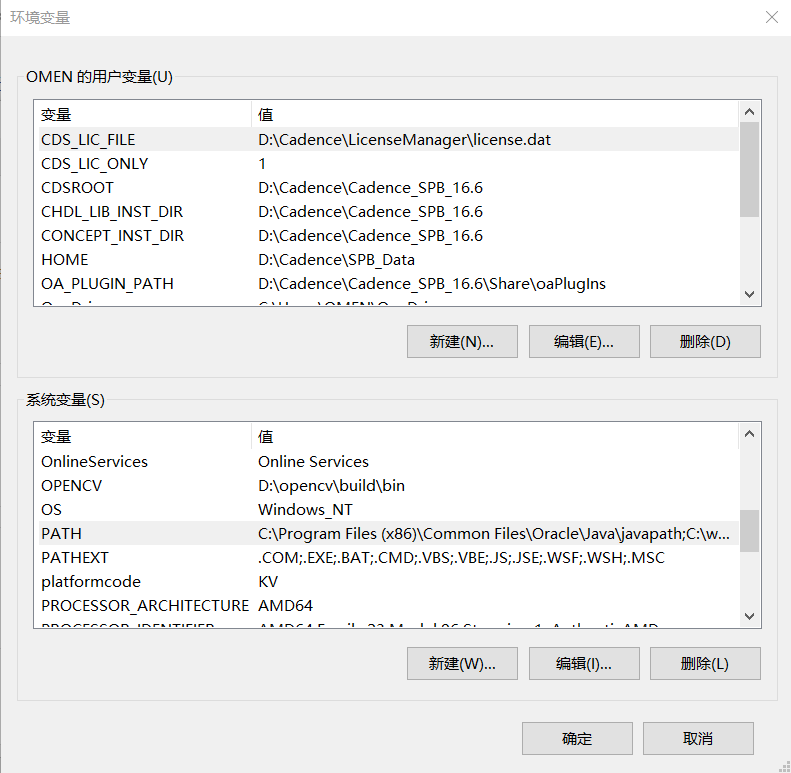
到官网<https://opencv.org/>下载最新Windows版的Opencv库



解压安装完成后，到指定的目录下寻找文件夹opencv，这就是库文件的存放处

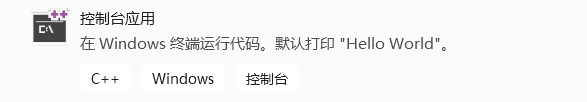
1. 配置系统变量

打开环境变量选项：此电脑（右键）——属性——高级系统设置——环境变量，在环境变量中的系统变量窗口找到PATH并点击编辑，然后新建条目，输入D:\opencv\build\x64\vc15\bin（一定要精确到bin文件夹，不能只选上层的文件夹）



1. 配置VS项目属性

新建控制台应用项目，使用C++编程



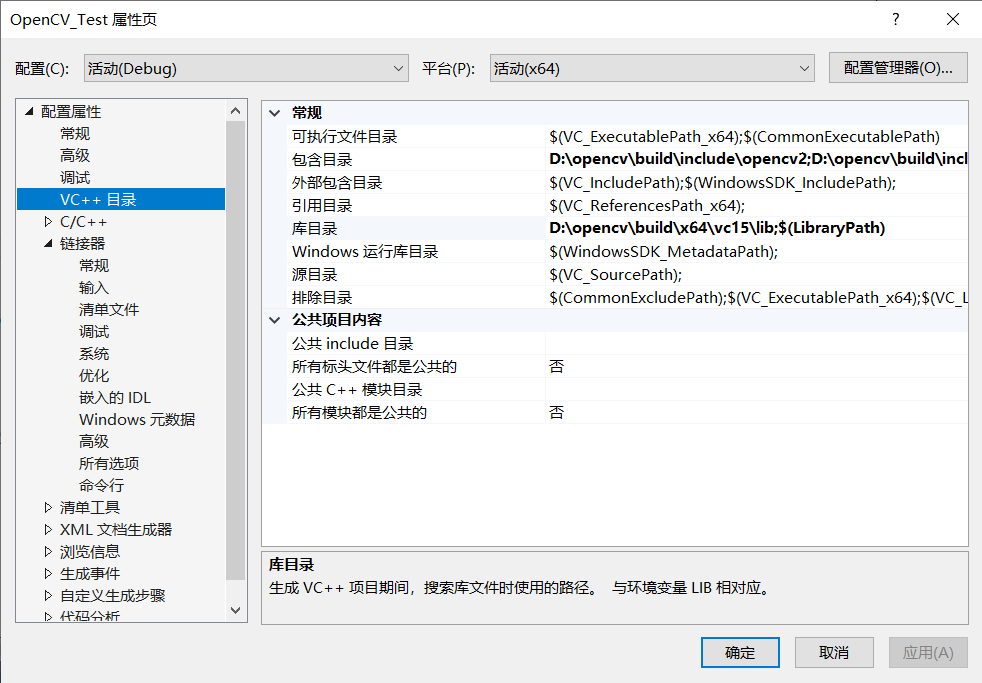
更改编译器：将调试栏处“Debug/Release”选项旁改成“x64”，使用64位编译器。



注意，这里选用Debug模式或Release模式均可，但后续的文件配置必须与之相配。

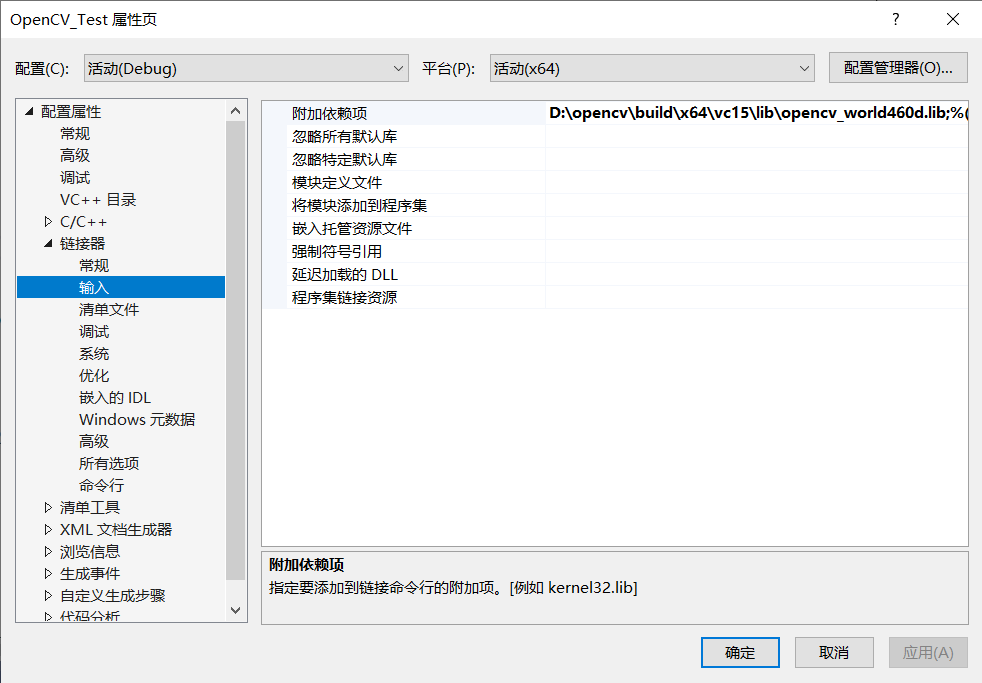
打开项目“属性”页，这里要修改三个地方：

1. VC++目录 / 包含目录：编辑，加入文件夹D:\opencv\build\include\opencv2和D:\opencv\build\include
2. VC++目录 / 库目录：编辑，加入文件夹D:\opencv\build\x64\vc15\lib



1. 链接器 / 输入 / 附加依赖项：编辑，加入文件夹D:\opencv\build\x64\vc15\lib\opencv\_world460d.lib（使用Debug模式）D:\opencv\build\x64\vc15\lib\opencv\_world460.lib（使用Release模式）

（460是版本号，会随着更新推出有所变动）



1. 代码引用头文件

最后，在代码中要添加#include引用Opencv库

|  |
| --- |
| #include <iostream>  #include <opencv2/core/core.hpp>  #include <opencv2/imgproc.hpp>  #include <opencv2/highgui/highgui.hpp>  using namespace std;  using namespace cv; |

## 函数Mat坐标参数顺序

Opencv中的Mat类代表一个矩阵，常用于储存图像。当用于储存图像时，Mat类中的一个元素即代表图像的一个像素。而每个像素坐标常有三种表示方式，xy坐标、行列（Rows-Cols）坐标和宽高坐标（Width-Height）。以图像最左上角的像素为原点，水平方向从左向右是x轴，x代表列（宽）；垂直从上到下是y轴，y代表行（高）。

容易发现，使用行列坐标时，第一个参数是实际是y值，第二个参数实际是x值。

一般情况下，涉及到Mat类访问像素的操作（如Mat.at<uchar>(y, x)）一般遵循数组的规范，使用行列坐标(y, x)；而绝大多数绘图函数的Point()使用xy坐标(x, y)，Size()函数使用的宽高坐标与(x, y)等价。

代码运行出错时，可以检查每个函数使用的坐标参数顺序是否正确。

## 从文件夹中批量读取文件

Opencv通常需要批量处理大量名字无规律的文件，glob()函数可以实现这个功能。函数glob()的作用是将指定路径文件夹中所有文件的目录存入一个字符串vector中。

|  |
| --- |
| #include <iostream>  #include <vector>  #include <opencv2/core/core.hpp>  #include <opencv2/imgproc.hpp>  #include <opencv2/highgui/highgui.hpp>  using namespace std;  using namespace cv;  string folder\_path = ".\\Test";  vector<string> allimage\_path; //用于存放文件地址的字符串向量  glob(folder\_path, allimage\_path, false); //文件夹Test中所有文件目录存入allimage\_path  cout << allimage.size() << endl; //文件数量  for (int i = 0; i < allimage.size(); i++)  {  cout << allimage[i] << endl; //各文件目录字符串  } |

# C++小技巧

## VS编译及调试相关

·代码批量注释及取消注释：Ctrl + K + /

·断点设置及取消：点击代码相应行最左侧灰色条

·跳入循环变量i=107时的内部结构：在循环中插入条件判断i==107?并设置断点

## 获取代码运行时间

函数clock()默认输出为时间单位为毫秒，除以1000换成秒。

|  |
| --- |
| #include <time.h>  clock\_t start, end;  start = clock();  /\*需要计时的代码\*/  end = clock();  cout << "Time = " << double(end - start) / CLOCKS\_PER\_SEC << "s" << endl; |

## 文件绝对路径与相对路径

绝对路径指的是从磁盘名开始逐级标识的文件路径，相对路径指的是以程序执行文件为参考位置标识的文件路径。一般为了保证程序的鲁棒性，应尽量使用相对路径。

|  |
| --- |
| string file = "D:\\Storage\\File\\data.txt"; //绝对路径  string file = ".\\Video\\v.mp4"; //"."表示当前目录的同级目录  string file = "..\\Image\\1.png"; //".."表示当前目录的上级目录 |

## 数值变量转换为字符串

函数to\_string()可以实现整形、浮点型等变量数值转换为字符串。另外，C++支持重载加号连接字符串。

|  |
| --- |
| int x = 10;  int y = 5;  string sign = "x = " + to\_string(x) + ", y = " + to\_string(y);  cout << sign << endl; //"x = 10, y = 5" |

## 字符串转换为数值变量

C语言的函数atoi()和atof()可分别用于将字符串转化为整形和浮点型数，它们的共同特点是从输入字符串的第一个合法字符（数字或负号）开始读取，直到遇到非法字符。atof()支持小数和科学计数法格式。

唯一问题是，这两个函数只接受字符指针，因此string类要调用成员函数c\_str()返回其常指针，不能直接输入。

|  |
| --- |
| #include <algorithm>  #include <string>  string str1 = "1234";  string str2 = "-123.456";  string str3 = "123e-4";  int data1 = atoi(str1.c\_str()); //整型  double data2 = atof(str2.c\_str()); //浮点型（小数）  double data3 = atof(str3.c\_str()); //浮点型（科学计数） |

## 循环结构的临时变量

在for循环判断体内部声明的循环变量i是临时变量，循环结束后即释放内存，下次使用时需重新声明。

|  |
| --- |
| for (int i = 0; i < 10; i++)  {  ...;  }  //此时i已被释放 |

## 编程规范相关

·循环变量最好从0开始，而且每次循环只加一，方便对数组的访问；

·程序运行中不改变的常量参数，出于调参的方便，在代码开头定义为宏变量；

# 问题记录