

# 视觉任务的指引

## 1. Visual Studio 2019 的安装

建议可以参考 b 站上面的一些教学安装视频，也可以上官网进行安装社区版。

官网地址：

[Visual Studio: 面向软件开发人员和 Teams 的 IDE 和代码编辑器 \(microsoft.com\)](https://visualstudio.microsoft.com/)

### 了解 Visual Studio



Community 为社区免费版

Professional 为专业的付费版

我们用来学习利用免费版即可。

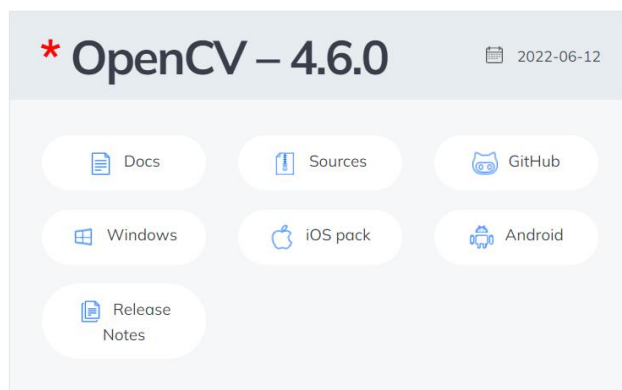
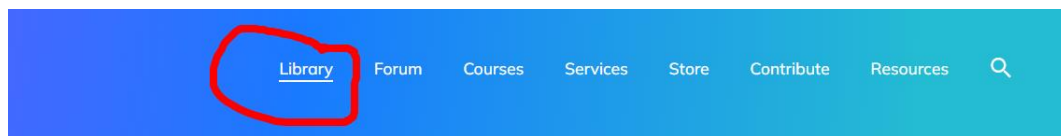
## 2. OpenCV4 的环境配置

安装好 Visual Studio 后开始准备安装我们用来处理图像的工具包即 OpenCV。

我现在安装是 OpenCV4.6.0。大家要主要 VS 的版本是跟 OpenCV 的版本要相匹配的，要是 OpenCV 的版本过新但是 VS 的版本过低会导致不兼容。

## 2.1 在 window 系统下安装 OpenCV 的主要步骤

首先在 OpenCV 的官网首页中找到“Library”选项并进入，里面会有历史发布的 OpenCV 各个版本，根据自己安装的 VS 版本寻找适合兼容的 OpenCV 版本并且下载（我用来举例的是 VS2019,并且安装的 OpenCV4.6.0 版本）。

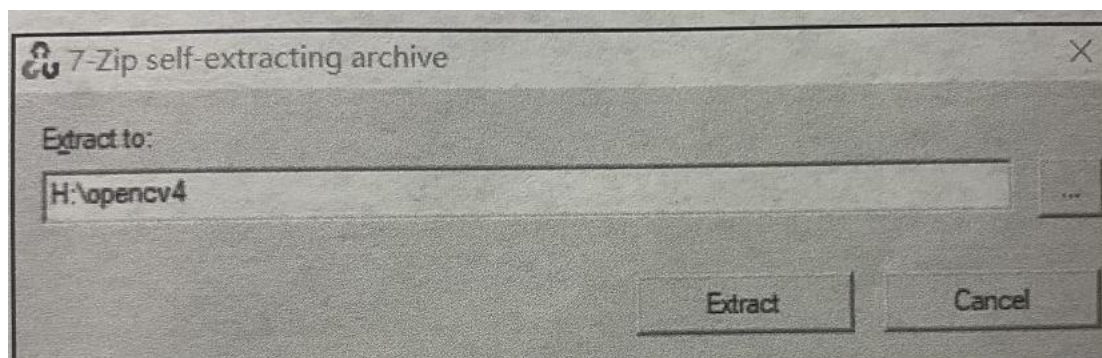


下载完成后会得到一个这个样的文件

 opencv-4.6.0-vc14_vc15.exe	2022/8/1 18:36	应用程序	231,176 KB
 Mall Customers (1).csv	2022/7/29 20:15	YIS 工作表	4 KB

点击这个文件即可开始安装 OpenCV 的包。

为了以后的处理建议在自己的 D 盘创建一个 Opecv4 的文件夹，用来装后续安装的文件。



选择好路径之后点击下面的[Extract]即可进行安装。  
安装完后要检查一下之前选择的文件夹下面是否多了名为 OpenCV 的文件夹。

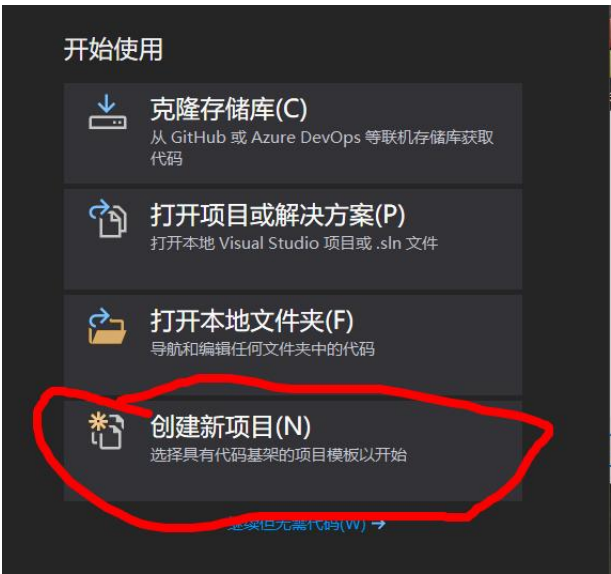
名称	修改日期	类型
opencv	2022/8/1 18:37	文件夹

点开里面是这样的

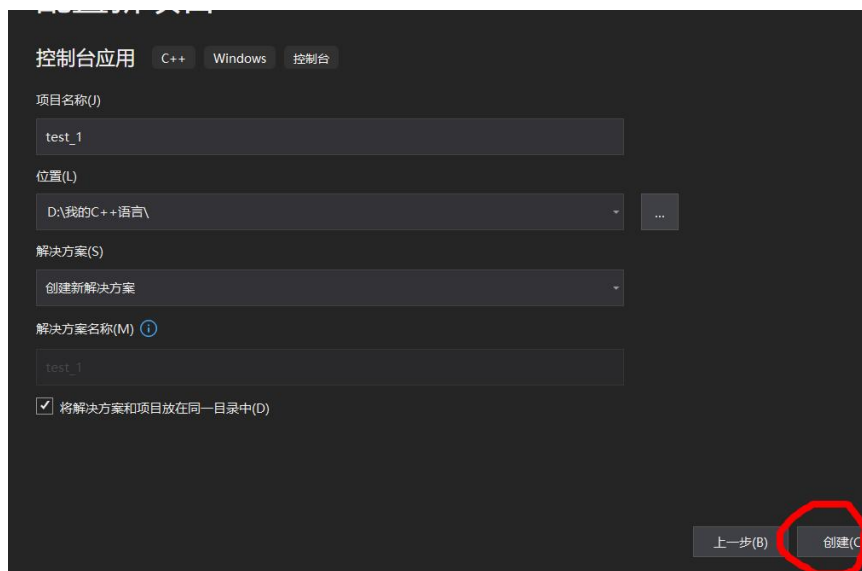
名称	修改日期	类型	大小
build	2022/8/1 18:37	文件夹	
sources	2022/8/1 18:37	文件夹	
LICENSE.txt	2022/6/6 0:09	文本文档	12 KB
LICENSE_FFMPEG.txt	2022/6/6 0:09	文本文档	28 KB
README.md.txt	2022/6/6 0:09	文本文档	1 KB

## 2.2 配置包括路径

配置环境的目的是为了告诉计算机安装的 OpenCV 在哪里。  
下面启动 VS（示范为 VS2019 社区版）

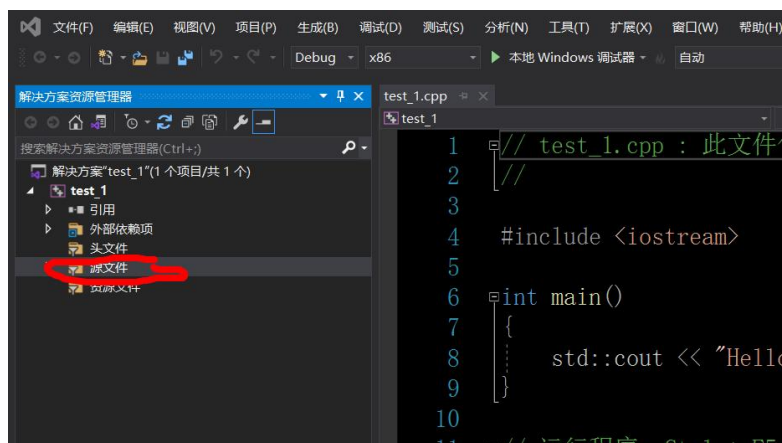


因为我们是测试 OpenCV,所以我们取名字为 test。



把控制台建立好后，可以看到左边弹出的解决方案管理器。

然后右击源文件开始建立 **cpp** 文件。

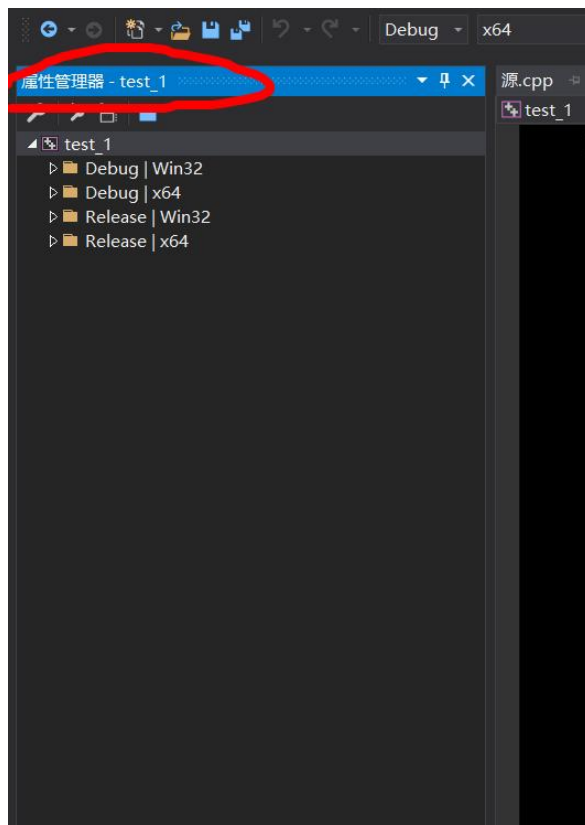




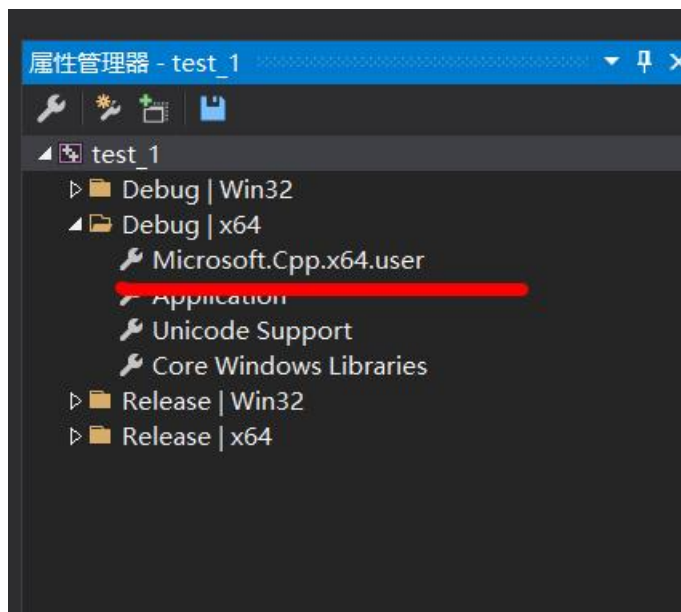
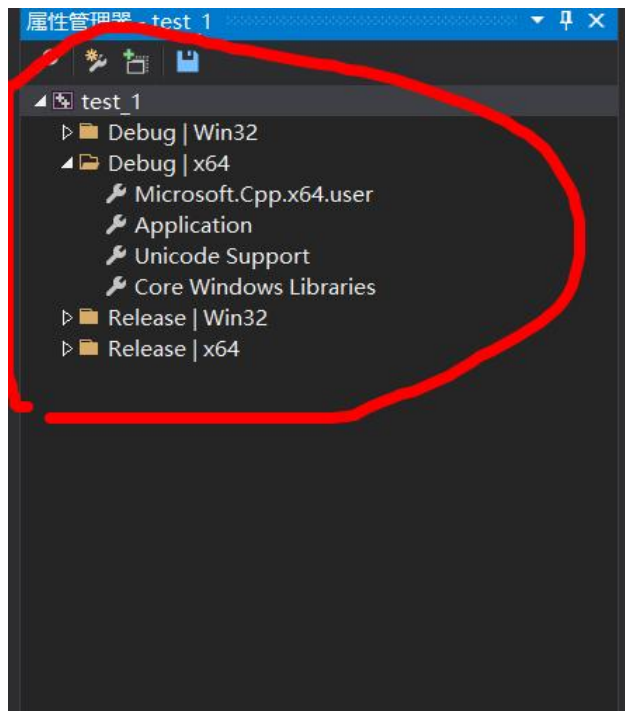
在完成空项目的建立后，就可以正式开始路径的配置，  
一次单机界面上方的搜索



搜索出属性管理器。

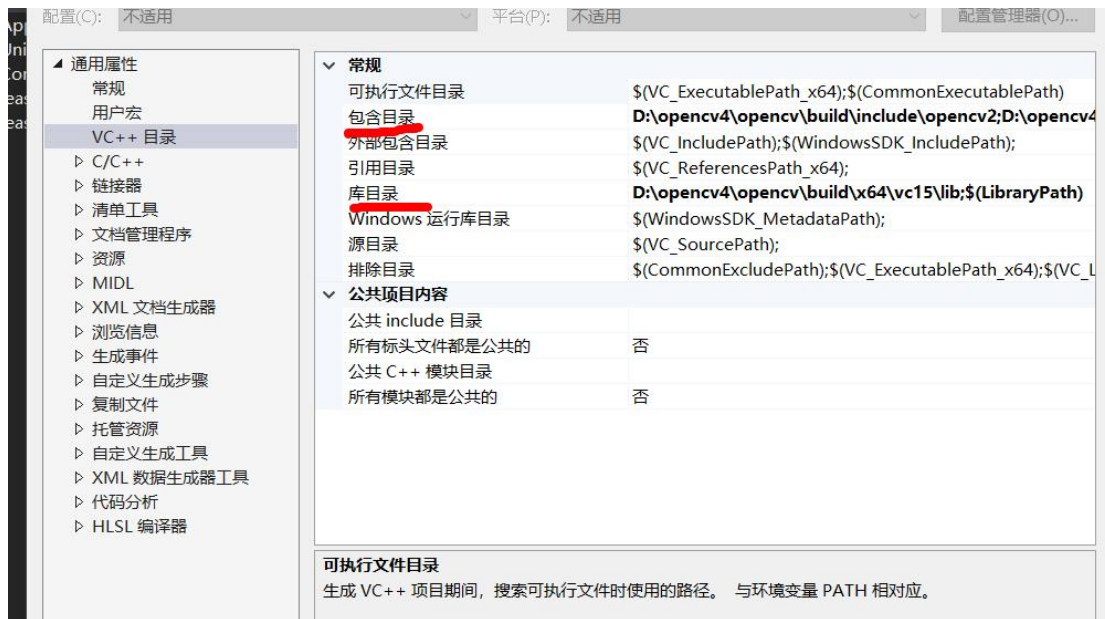


点开 Debug ×64.

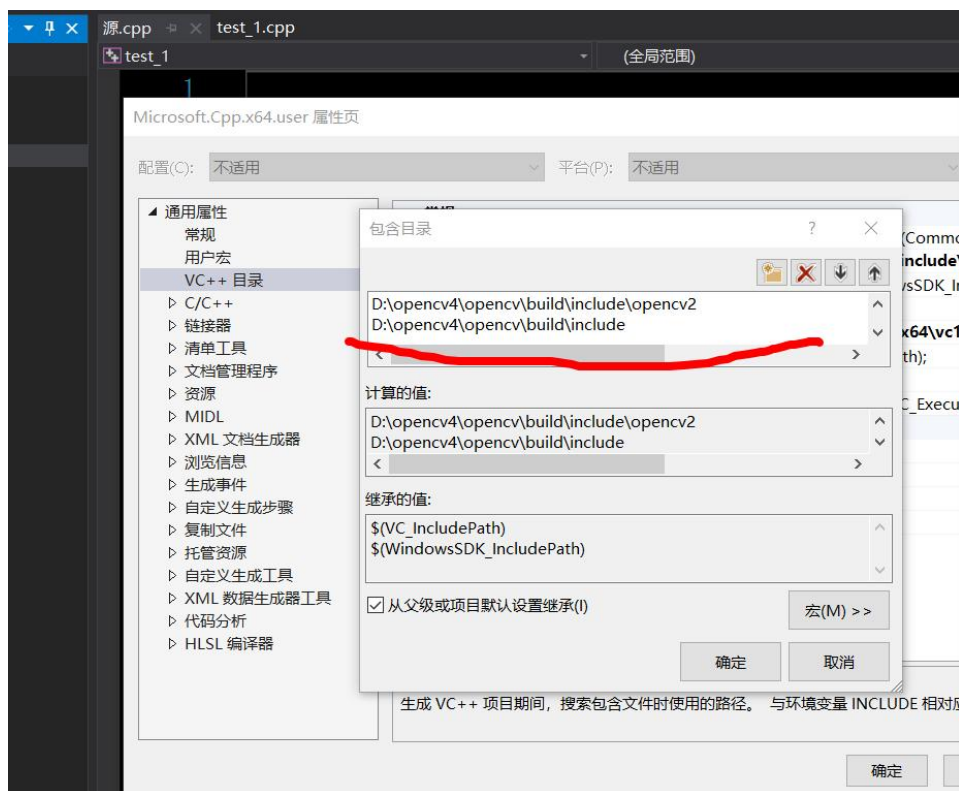


这个里面的 VC++ 目录的包含目录和库目录  
就是我们要修改的对象。





在包含目录里面添加两个文件的路径，图中是我已经添加好的情况。

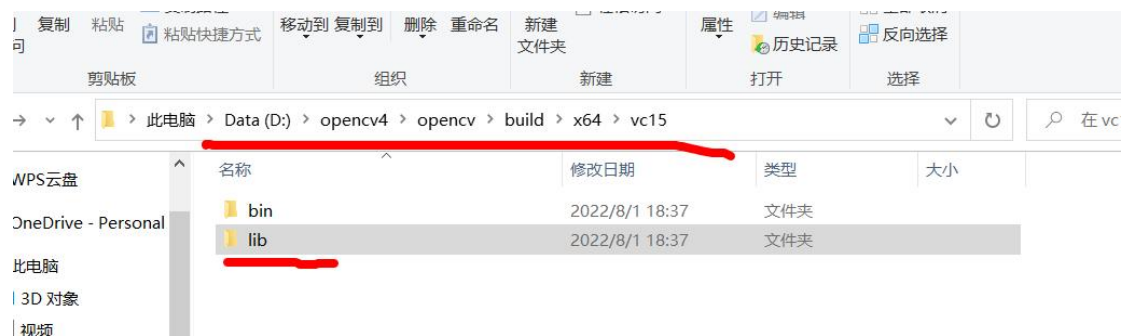
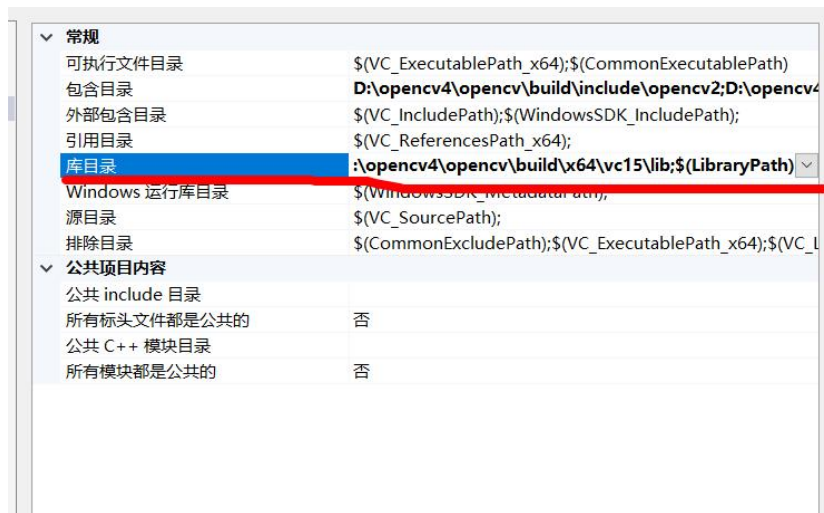


这是原本安装好的文件的路径复制粘贴好在这里

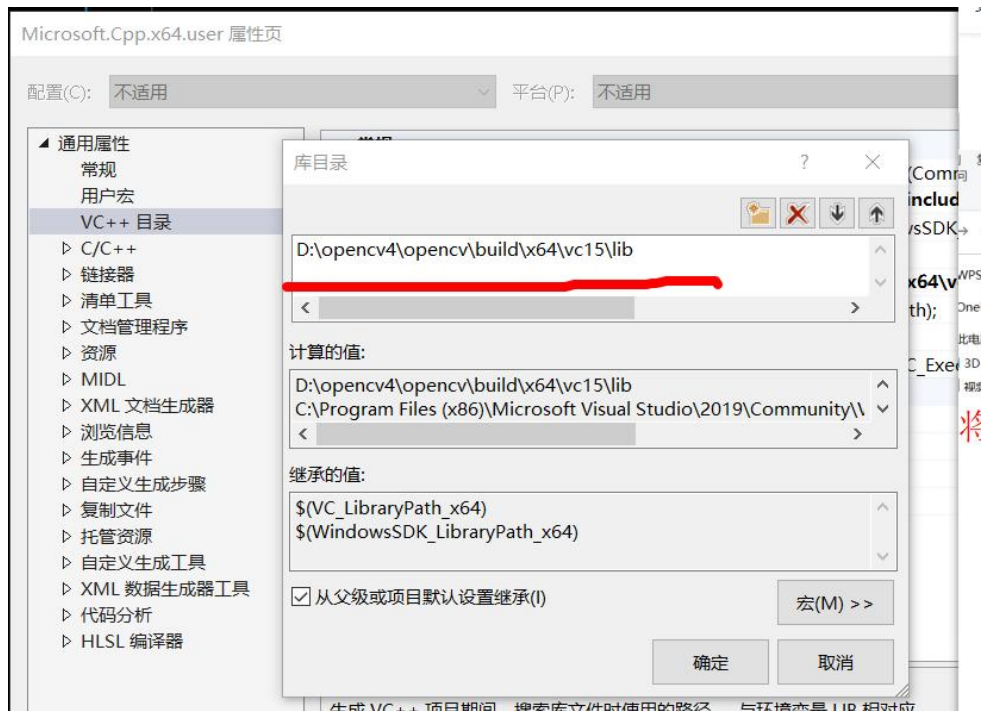
在里面添加好自己之前安装 OpenCV 的路径，图中是我安装的路径，每个人的路径不同，请注意！！！！



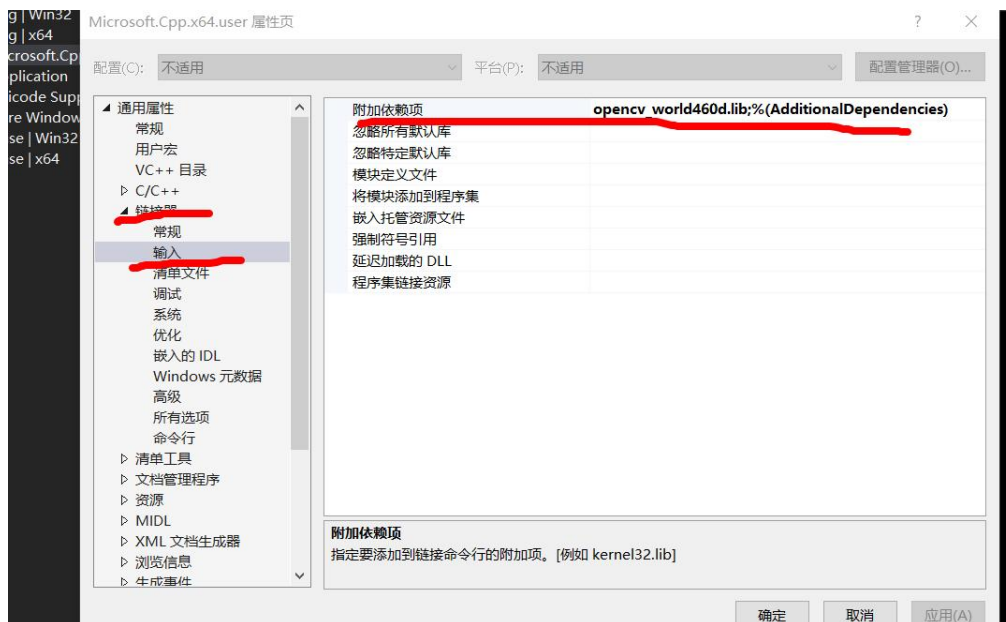
## 2.3 库目录的配置



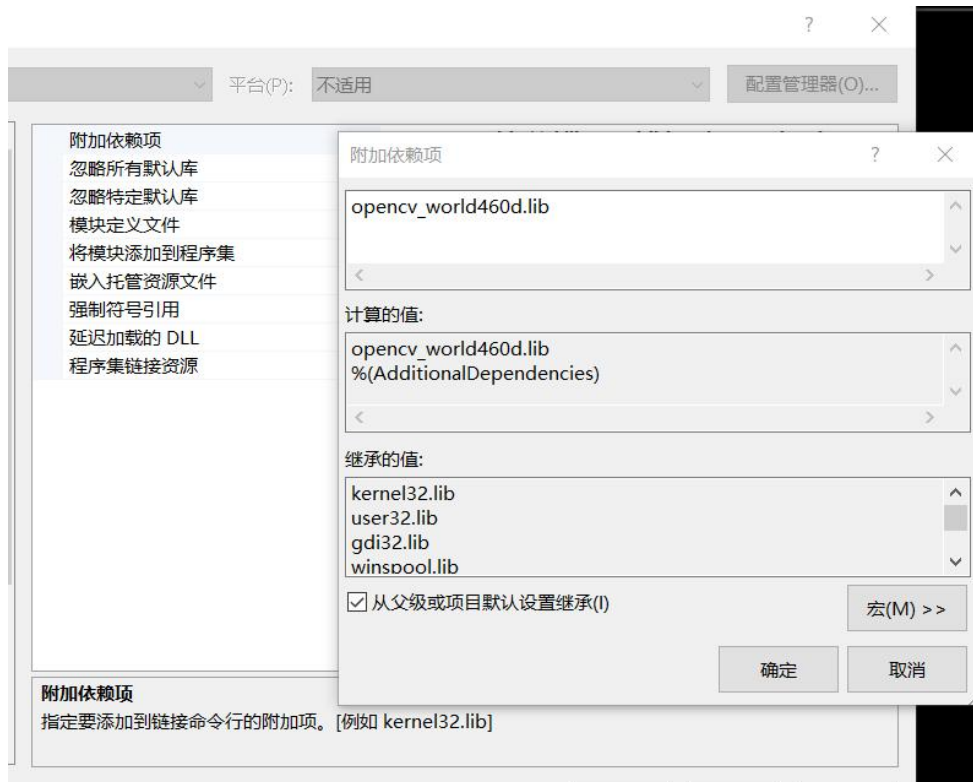
将这个文件夹的路径复制粘贴到库目录里面。



## 2.4 配置链接器



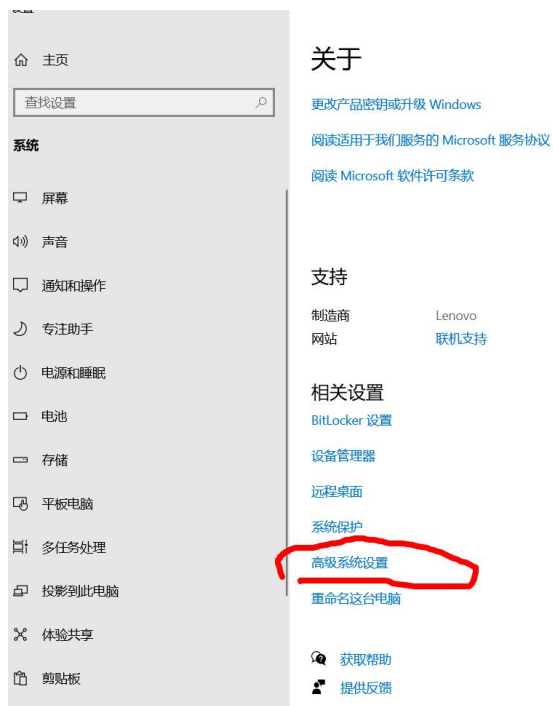
在附加依赖项里面添加



完成连接器的配置

## 2.5 完成系统变量的配置

在“我的电脑”右击，进入属性



找到这个高级系统设置，然后点击进入环境变量。



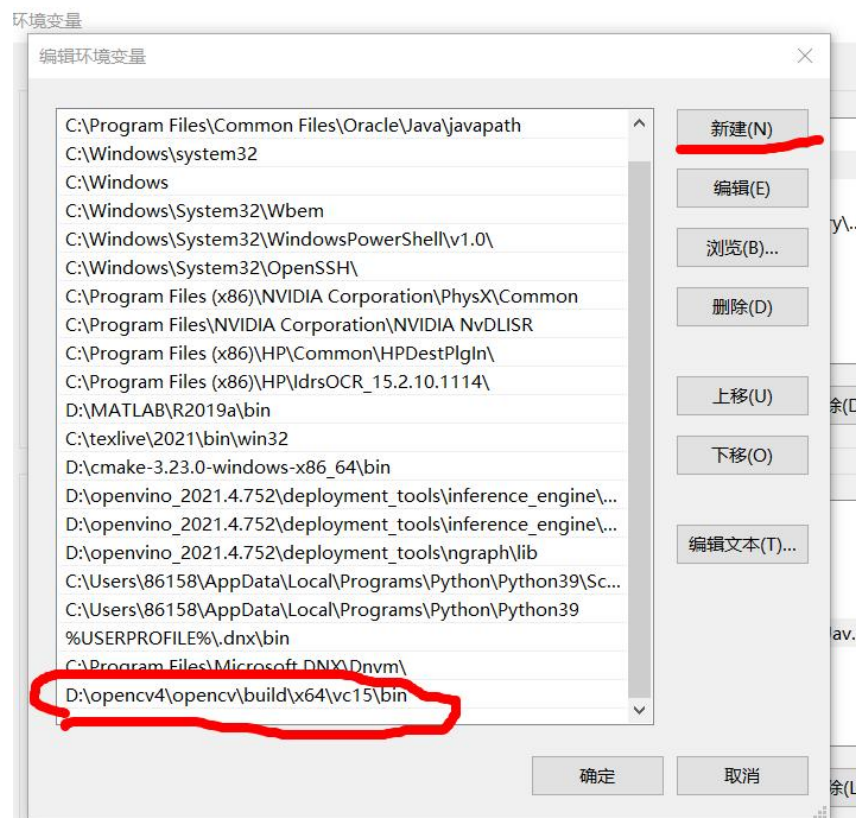
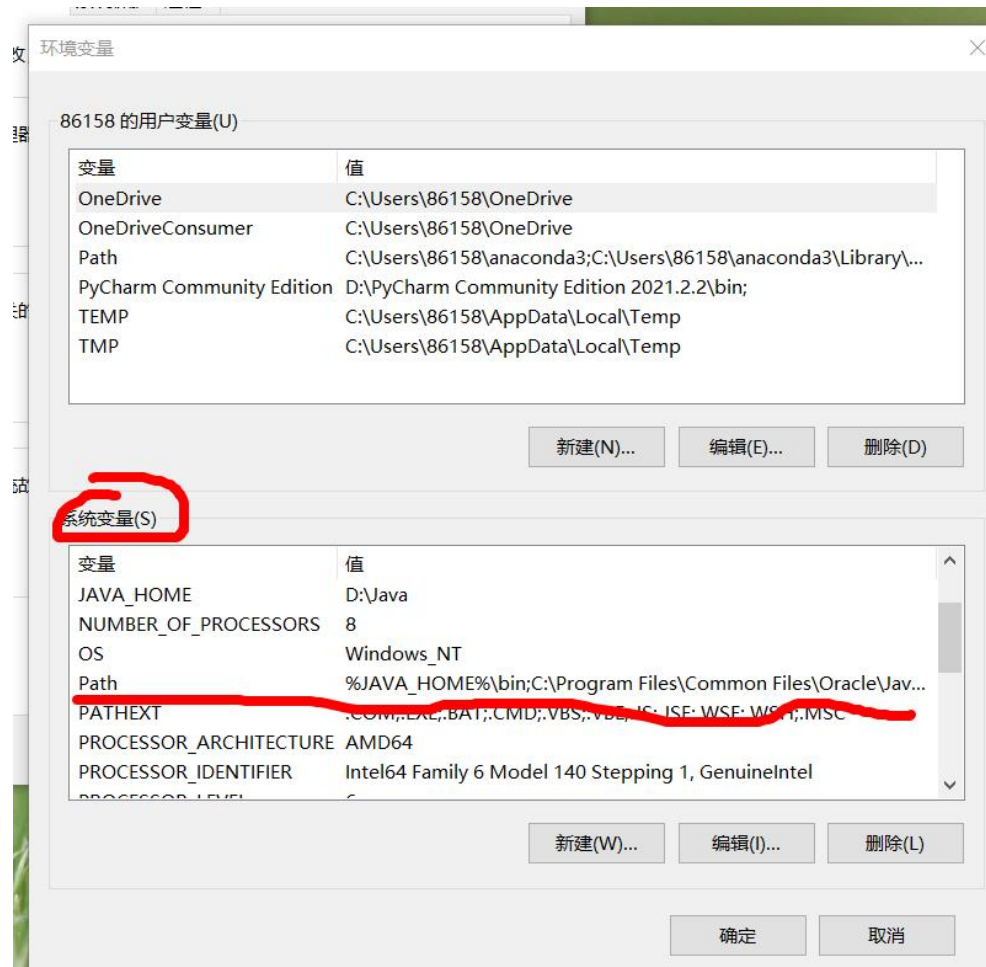
找到这个系统变量里面的 Path

系统变量！！！！

系统变量！！！！

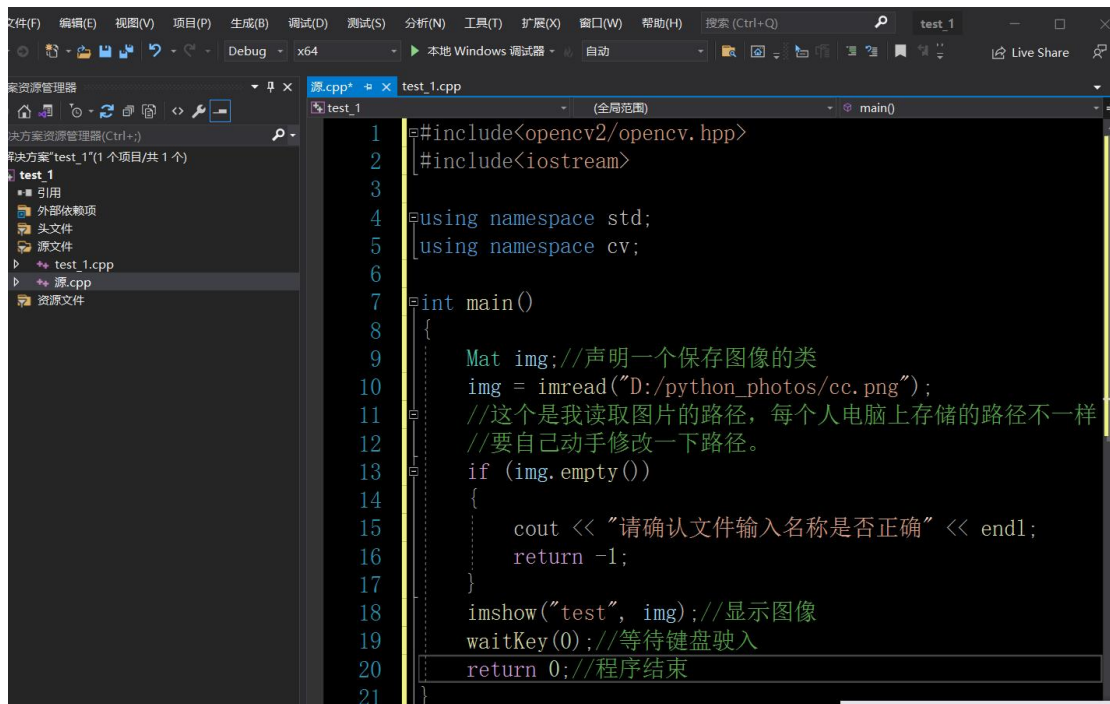
系统变量！！！！

重要的事说三遍。双击进去！！



自己新建一个路径把这个 bin 文件装进去。

### 3. 验证是否配置完毕。



```
1 #include<opencv2/opencv.hpp>
2 #include<iostream>
3
4 using namespace std;
5 using namespace cv;
6
7 int main()
8 {
9     Mat img; //声明一个保存图像的类型
10    img = imread("D:/python_photos/cc.png");
11    //这个是我读取图片的路径，每个人电脑上存储的路径不一样
12    //要自己动手修改一下路径。
13    if (img.empty())
14    {
15        cout << "请确认文件输入名称是否正确" << endl;
16        return -1;
17    }
18    imshow("test", img); //显示图像
19    waitKey(0); //等待键盘输入
20    return 0; //程序结束
21 }
```

如果一开始两个 include 文件头没有报错即可确认 OpenCV 的包已经导入成功。

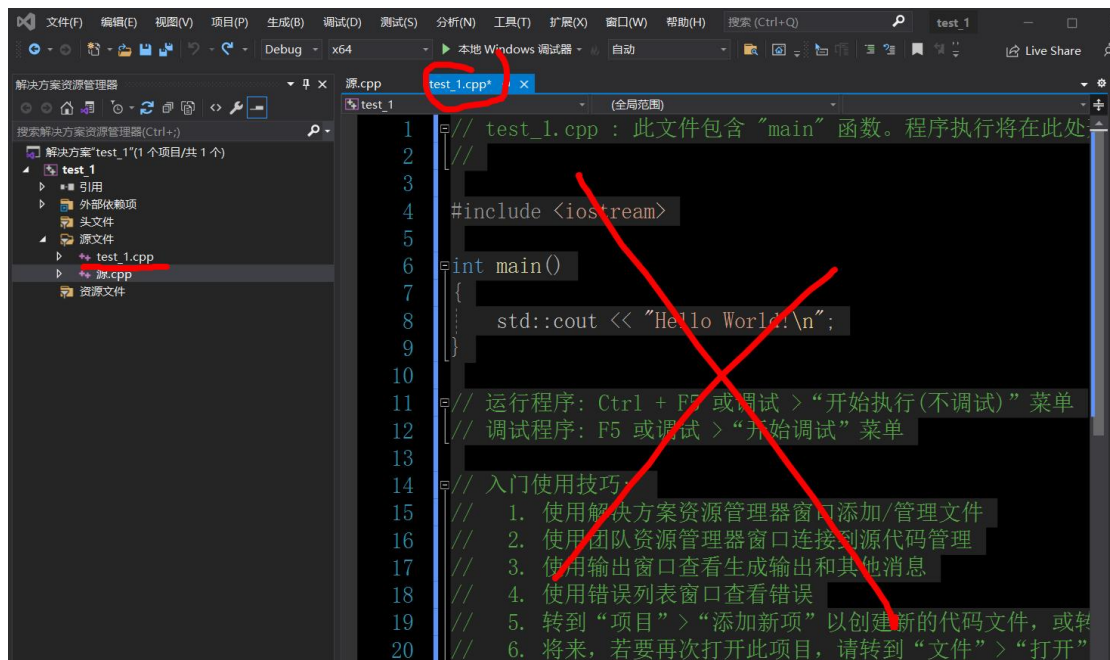
想动手的同学可以自己在一开始创建的 cpp 文件里面进行手打。

不想手打的可以复制粘贴下面的代码验证



(温馨提示)

因为我自己尝试的时候出现一个报错，  
因为我这个项目里面有两个 **cpp** 文件，其中另一个已经存在了一个 **main** 函数，会导致运行错误，所以将其中一个 **cpp** 文件清空即可运行。



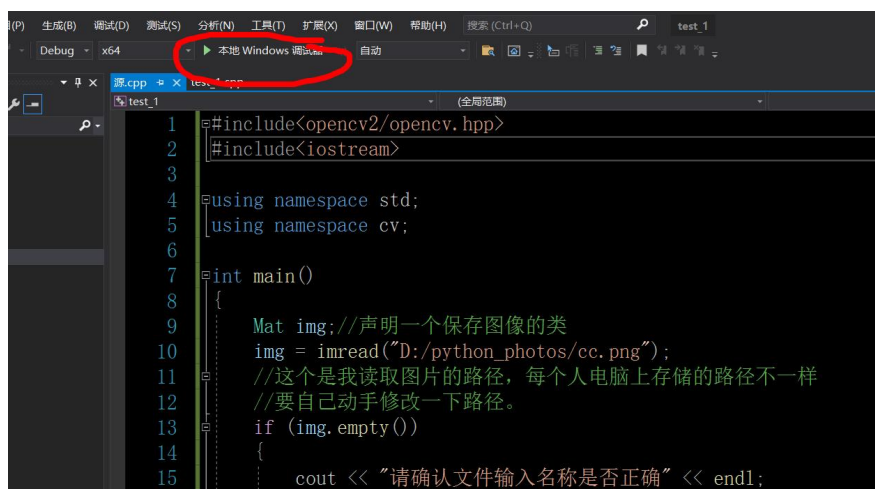
但是！！！！要自己准备好要打开的图片，并且对下面代码中图片的路径进行修改。

```
#include<opencv2/opencv.hpp>
#include<iostream>

using namespace std;
using namespace cv;

int main()
{
    Mat img;//声明一个保存图像类
    img = imread("D:/python_photos/cc.png");
    //这个是我读取图片的路径，每个人电脑上存储的路径不一样
    //要自己动手修改一下路径。
    if (img.empty())
    {
        cout << "请确认文件输入名称是否正确" << endl;
        return -1;
    }
    imshow("test", img);//显示图像
    waitKey(0);//等待键盘输入
    return 0;//程序结束
}
```

点击图片位置运行代码



执行出来的效果如下所示！！！！



The screenshot shows a C++ IDE with a project named 'test\_1'. The main window displays a meme featuring a cartoon character with a topknot and a fan, with the text '我书读得多，不会骗你' (I read a lot of books, I won't lie to you). The background shows the output console with various log messages, including 'INFO:000.084' and 'global c:\build\master\_winpack-build-win64-vc15\opencv\modules\core\src\utils\plugin\_loader.impl.hpp (67)'. The code editor shows the following code:

```
15 cout << "请确认文件输入名称是否正确" <<  
16 return -1;  
17 }  
18 imshow("test", img); //显示图像  
19 waitKey(0); //等待键盘输入  
20 return 0; //程序结束  
21 }  
22
```