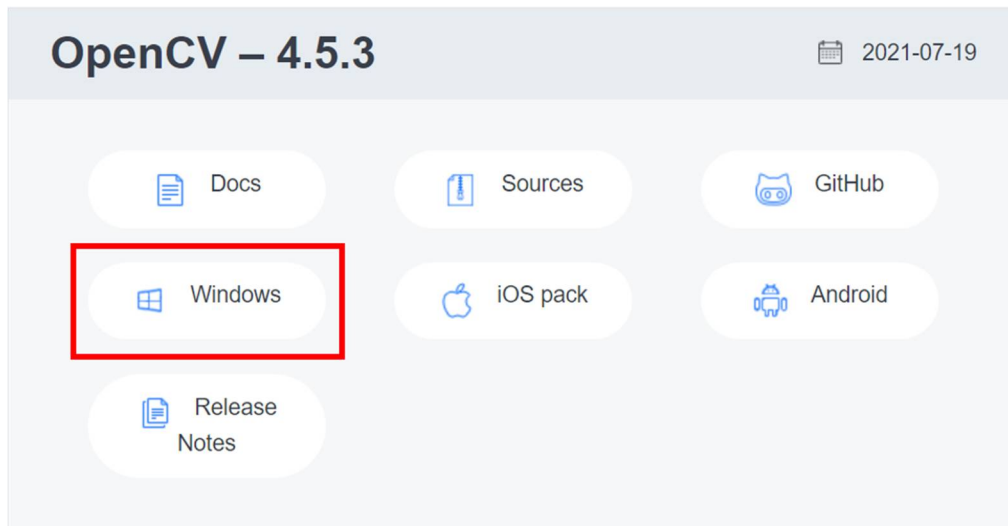


## OpenCV 安裝 (Windows 系統版本)

\*在此以 4.5.3 為範例做安裝

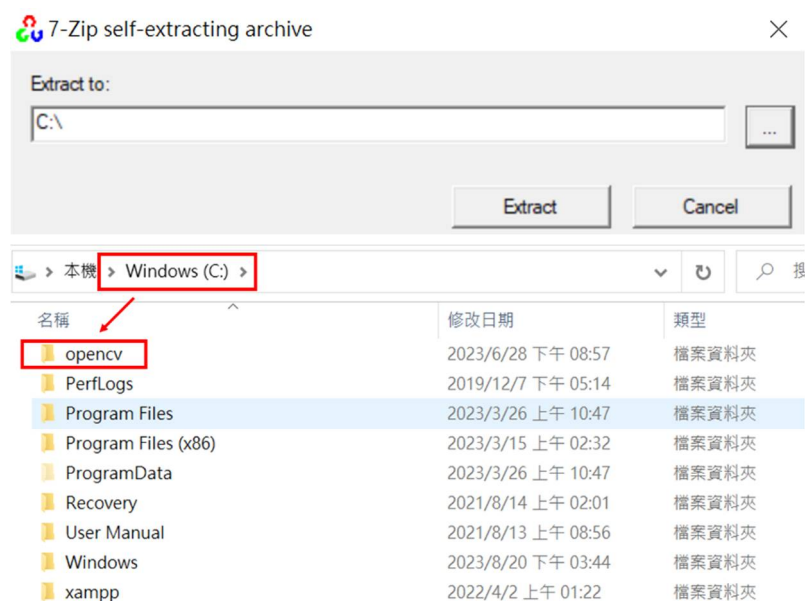
### 1. 進入官網，下載軟件（網址：<https://opencv.org/releases/>）

選擇 4.5.3 的版本，並點擊 Windows 之選項，進行下載，  
隨即會跳轉至下載頁面，等待幾分鐘後即下載完成。



### 2. 執行 exe 檔，並將檔案存至 c 槽

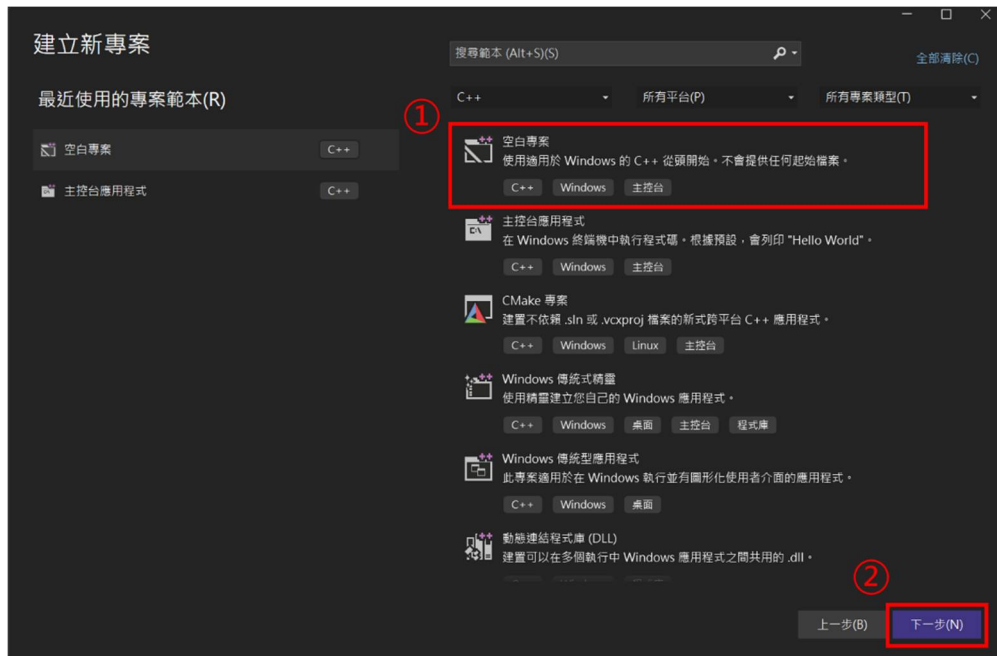
執行下載完畢的 exe 檔，將路徑選擇在 C:\，  
經過一系列流程後，即可獲得 OpenCV 的資料夾。



### 3. 建立 OpenCV 專案

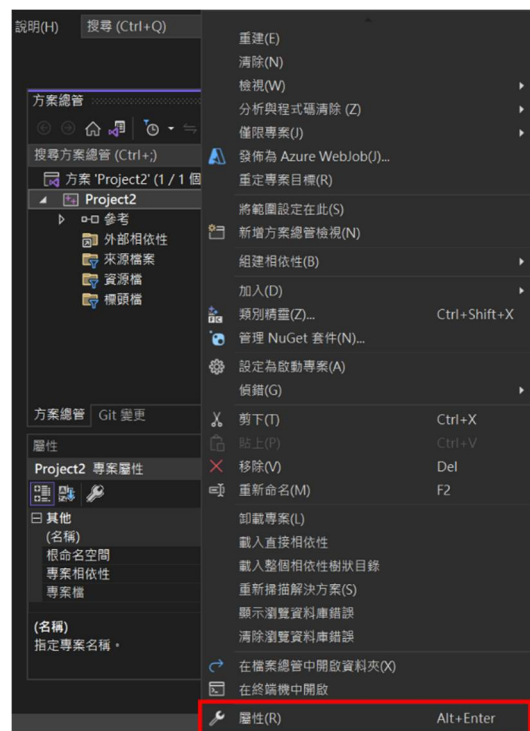
\*在此使用 VS 2022 為範例操作

開啟 VS 2022，選擇 C++ 空白專案(建議存在一個固定處，如此一來所有的設定就能延用於後續的作業，不同作業僅需開啟新的 cpp 檔即可)。



### 4. 設定專案屬性之內容

開啟專案後，於方案總管的專案(此圖片為 Project2)上按右鍵，選擇「屬性」。

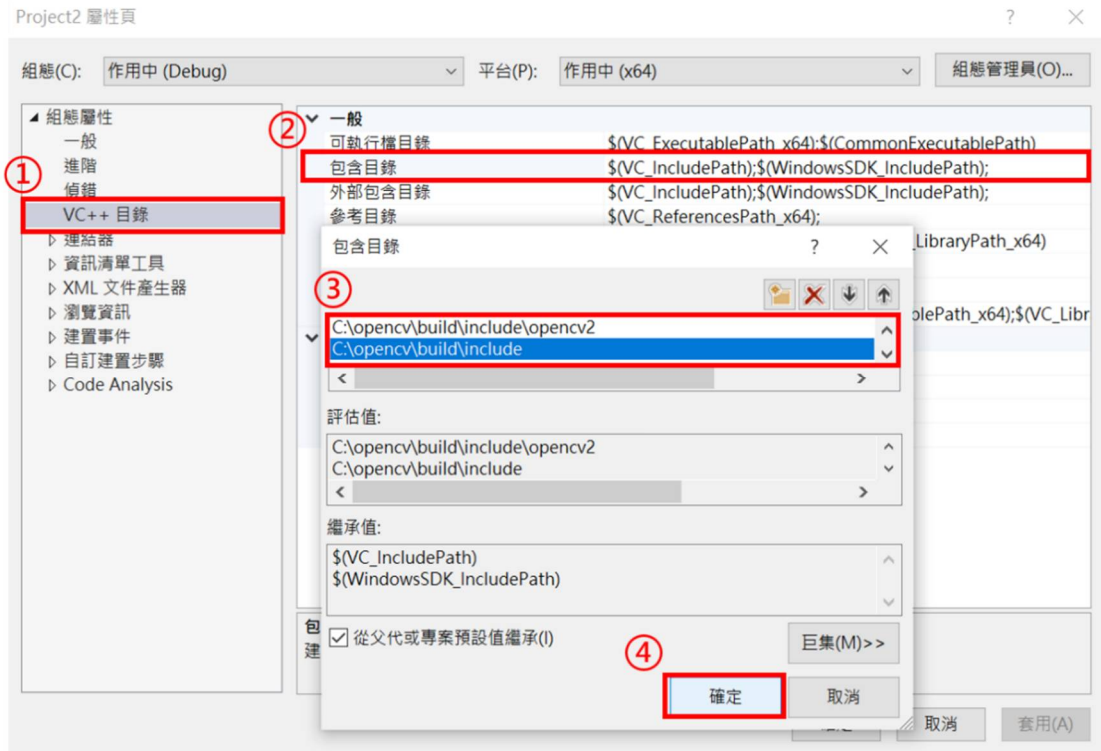


## 5. 加入檔案

在 VC++ 目錄 → 包含目錄(include 目錄) → 編輯 · 加入 OpenCV 的檔案路徑。

C:\opencv\build\include\opencv2

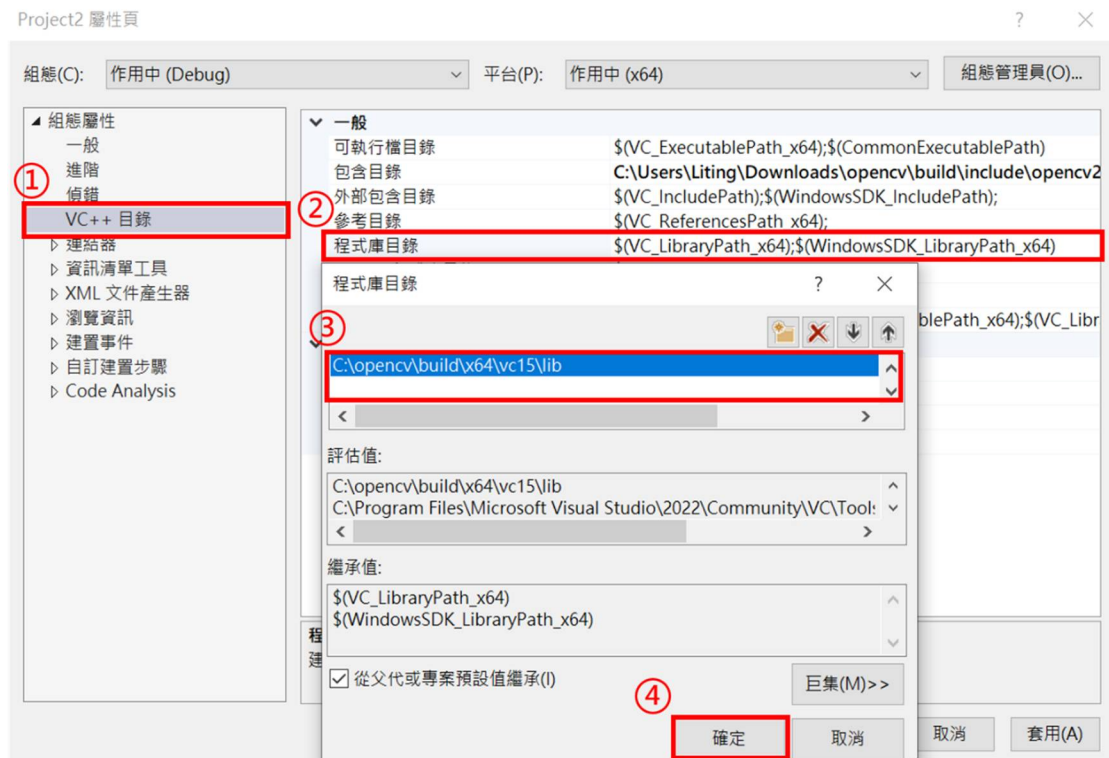
C:\opencv\build\include



## 6. 加入函式庫

在 VC++ 目錄 → 程式庫目錄 → 編輯，加入 OpenCV 的函式庫路徑。

C:\opencv\build\x64\vc15\lib

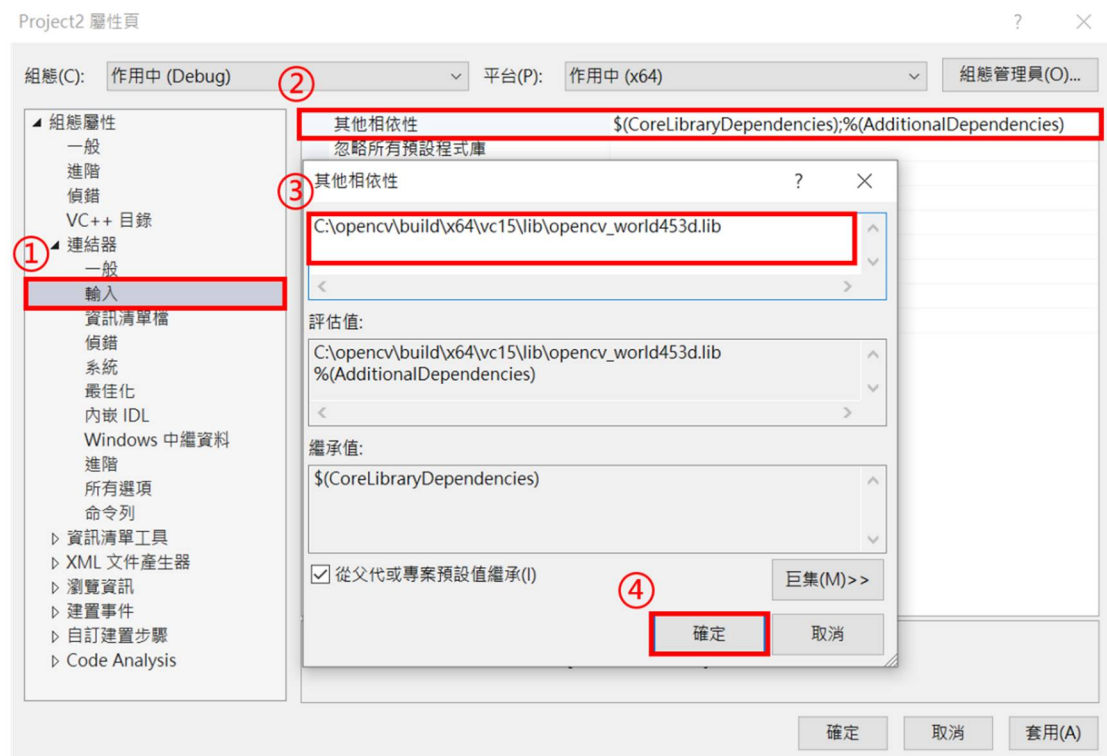


## 7. 加入相依性檔案

連結器(Linker) → 輸入 → 其他相依性 → 編輯 · 加入 lib 檔。

\*打開資料夾會發現有 opencv\_world453d.lib 和 opencv\_world453.lib 兩種，其中 d 代表 debug 模式，而沒有 d 代表 release 模式。

C:\opencv\build\x64\vc15\lib\opencv\_world453d.lib



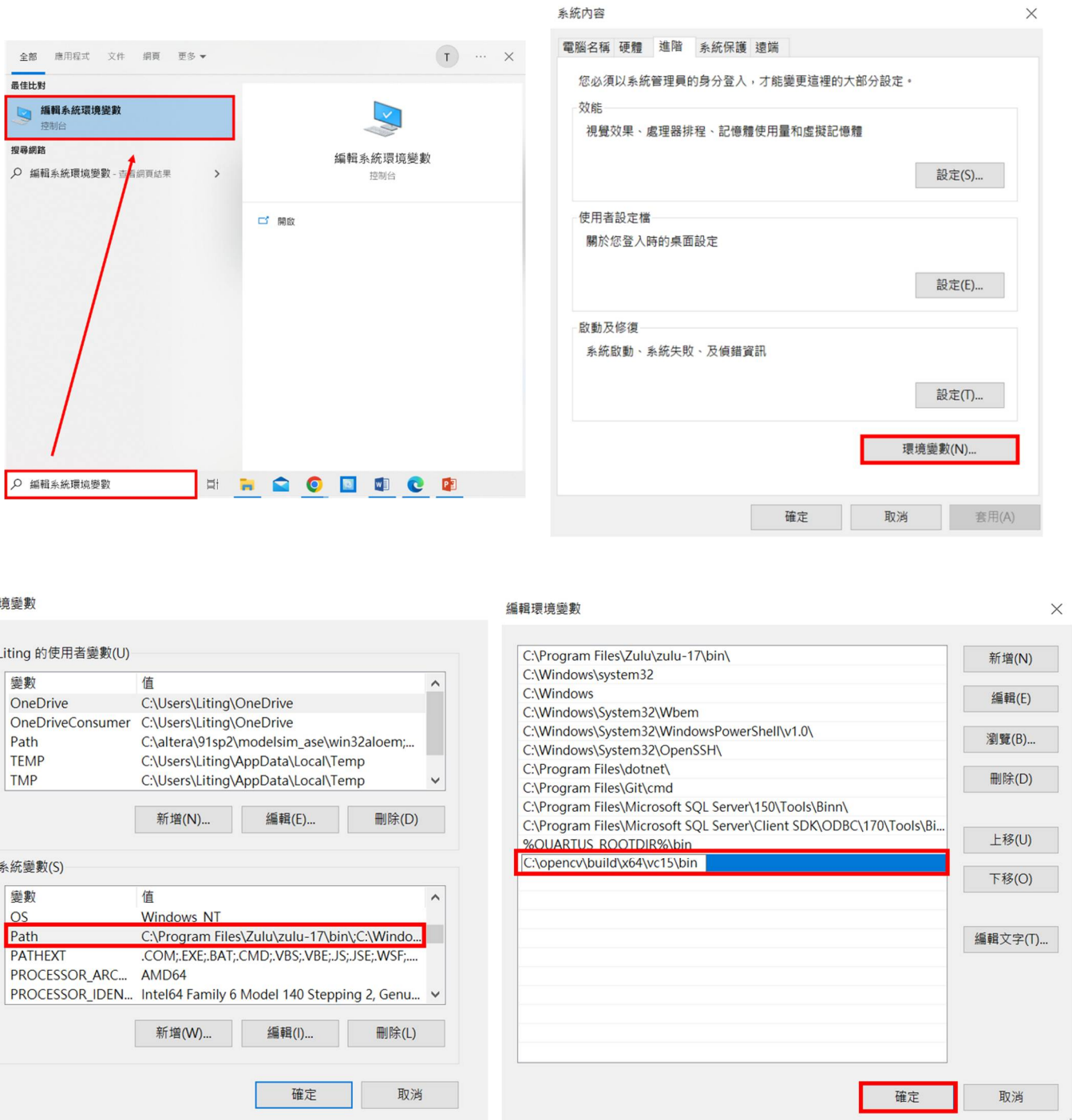
## 8. 加入環境變數

搜尋「編輯系統環境變數」→ 進階 → 環境變數，在系統變數的 Path 加入環境變數值。

\*按下確定後，關閉環境變數，並重開機讓系統變數生效。

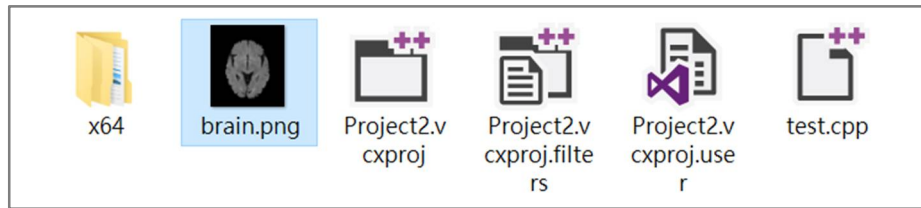
\*記得儲存專案！！！！

```
C:\opencv\build\x64\vc15\bin
```



## 9. 測試專案

重新開啟專案後，新增一個 cpp 檔 ( 在此以 test.cpp 為例 )，將一張圖片 ( brain.png ) 放入專案資料夾，與 cpp 檔儲存位子一致。



之後在 cpp 內寫入以下程式碼。

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

using namespace std;
using namespace cv;

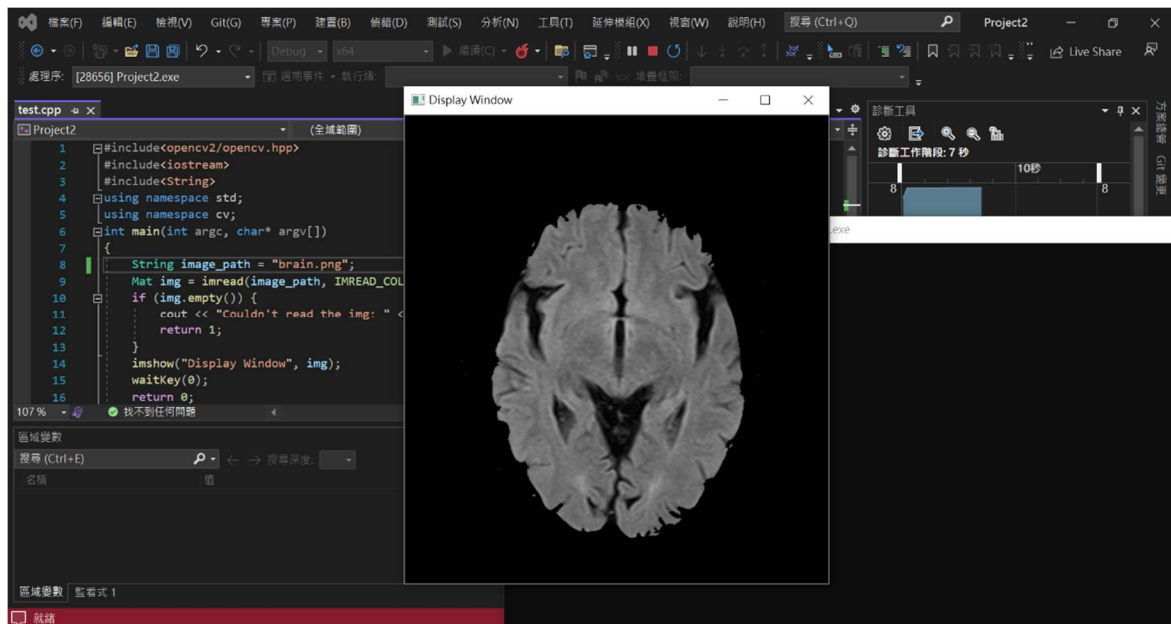
int main(int argc, char* argv[])
{
    String image_path = "brain.png";
    Mat img = imread(image_path, IMREAD_COLOR);
    if (img.empty()) {
        cout << "Couldn't read the img: " << image_path << endl;
        return 1;
    }
    imshow("Display Window", img);
    waitKey(0);
    return 0;
}
```

**提醒：**

1. 圖片檔案名稱請自行更改成相對應檔名
2. 透過 imread 將圖片導入
3. 透過 imshow 將圖片顯示
4. 透過 waitKey 將圖片停留

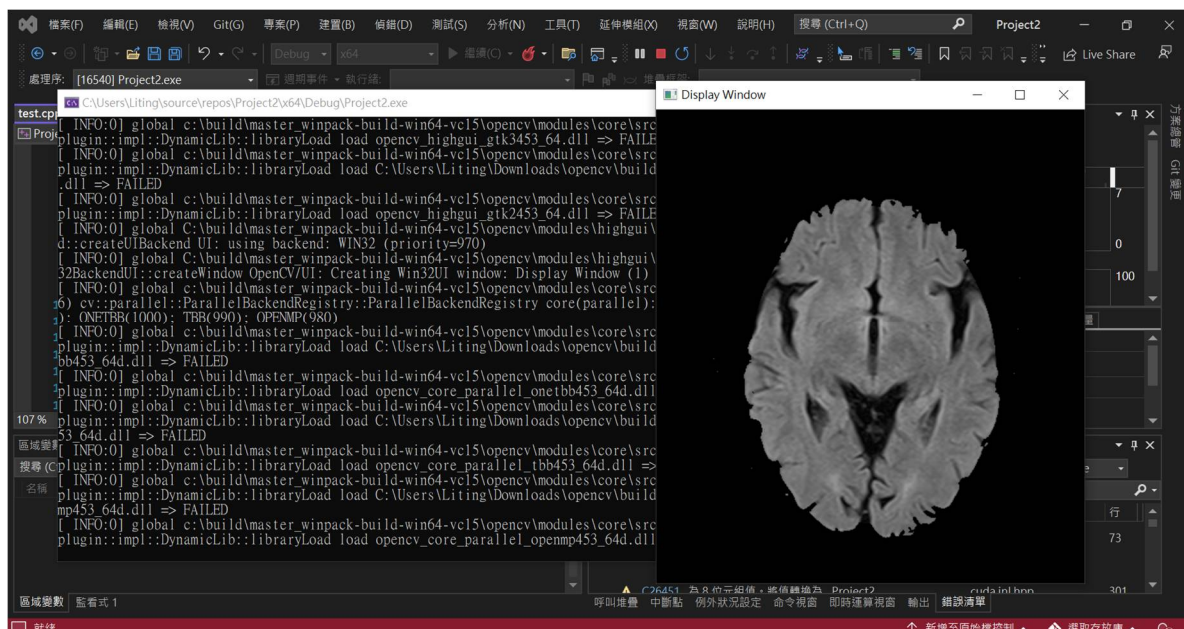


若能夠順利編譯並執行，則會出現以下畫面。



提醒：

如果出現以下情況可直接忽略，此訊息的等級只是 INFO，還不算是錯誤，只是系統顯示給使用者看，讓我們知道而已。





## 10. 在 Visual Studio 使用命令引數

在執行 C++ 程式時，我們其實可以透過指令傳遞參數給 main function，稱為命令列引數 (VS 中稱作為命令引數)。使用命令列引數可以避免每次執行時，都需要手動更改程式碼中的某些變數值。

### • Command line interface 介紹及基本指令

GUI ( Graphical User Interface )，圖形使用者介面，指通過圖形進行操作的使用者界面。

CLI ( Command-line Interface )，命令行界面，使用者通過鍵盤輸入指令，電腦接收到指令後，予以執行。

GUI 動作	CLI 指令( Mac OS / Linux )	CLI 指令 ( Windows )
顯示檔案內容	cat	more, type
列出資料夾內所有檔案	ls	Dir
切換資料夾	cd	Cd
新增資料夾	mkdir	Mkdir
移動檔案	mv	Move
刪除檔案	rm	Del
執行可執行檔	./a.out	./a.exe.

※ 在 Linux 中，C++ 編譯後的可執行檔預設檔名是 a.out，Windows 則是 a.exe。

### • Compiler 編譯器

我們可將程式分為「編譯」和「執行」兩步驟。

如果編譯器為 g++，Linux 的編譯指令為：

```
g++ test.cpp -o test.out
```

編譯 test.cpp  
產生可執行檔

可執行檔命名  
為 test.out

執行指令則為：

```
./test.out arg1 arg2
```

而這整句都將會被傳進 main function 中，作為命令引數。第一個引數是“./test.out”，第二個引數則是“arg1”

- C++ 的命令引數

以前面專案為例，如想更改為命令引數的話，在 Linux 中會變為：`./test.out brain.png`。此時程式會從 `argv[0]` 讀得 `“./test.out”`，`argv[1]` 讀得 `“brain.png”`。而 `argc` 是一個整數，其值就是 `argv` 陣列的長度，在此為 2。

修改後的程式碼如下：

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

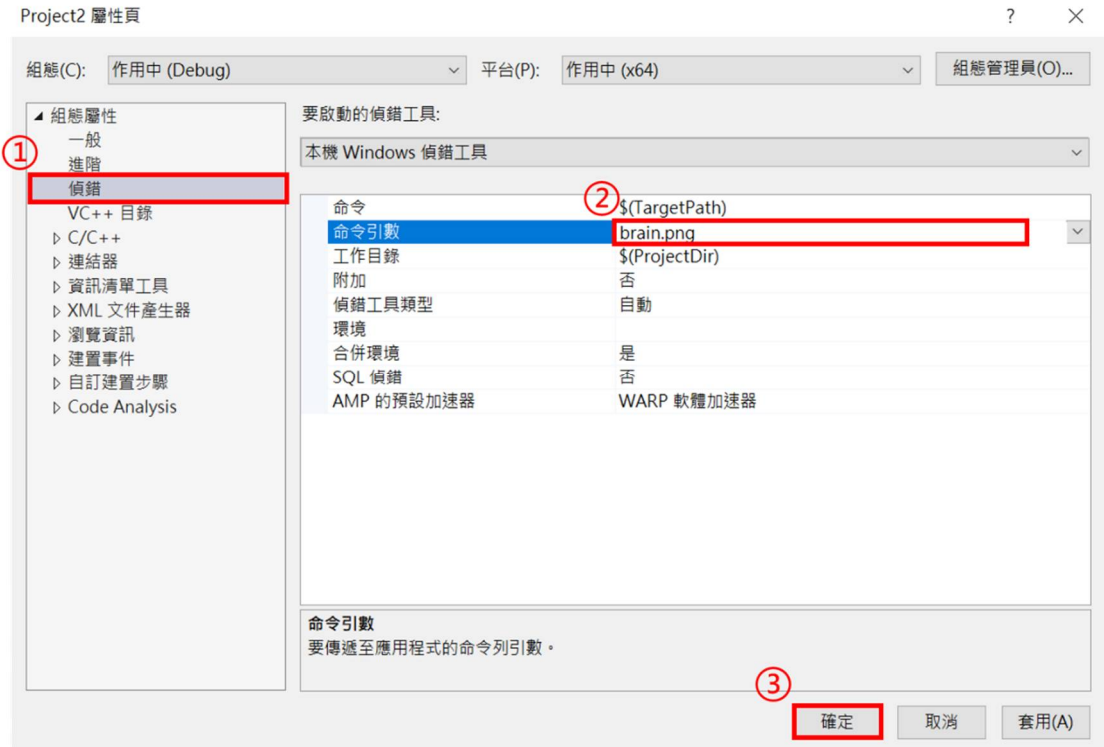
using namespace std;
using namespace cv;

int main(int argc, char* argv[])
{
    String image_path = argv[1];
    Mat img = imread(image_path, IMREAD_COLOR);
    if (img.empty()) {
        cout << "Couldn't read the img: " << image_path << endl;
        return 1;
    }
    imshow("Display Window", img);
    waitKey(0);
    return 0;
}
```

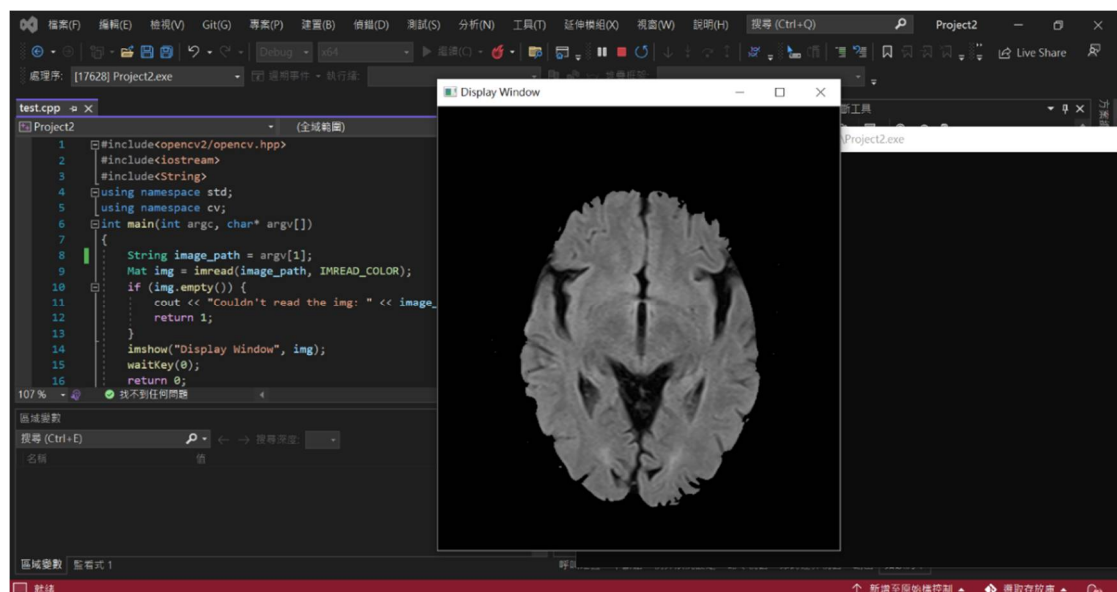
而 VS 屬性需更改：

專案 → 屬性 → 偵錯 → 命令引數

\*此時儘管圖片和 cpp 檔在不同資料夾中，也可以透過絕對路徑的方式獲得圖片。



執行結果如下：



## 11. 參考資料

[Visual Studio 2022 配置 Open CV](#)

[在 Visual Studio 中添加命令行參數 argv\[\]](#)

[main 函數讀取命令列參數，argc 與 argv 用法](#)

[command line interface 介紹及基本指令](#)

[command-line](#)

文：112 線代助教-羅立庭