

基于yolo的像素分割使用文档

1. 环境配置

支持的操作系统：Windows10/11 X64

此项目只支持在Nvidia显卡上运行。不支持AMD等其他品牌，否则只能使用cpu进行预测
如果只想使用cpu进行推测，不需要安装显卡驱动、cuda工具包、cudnn

1. 使用cuda的依赖

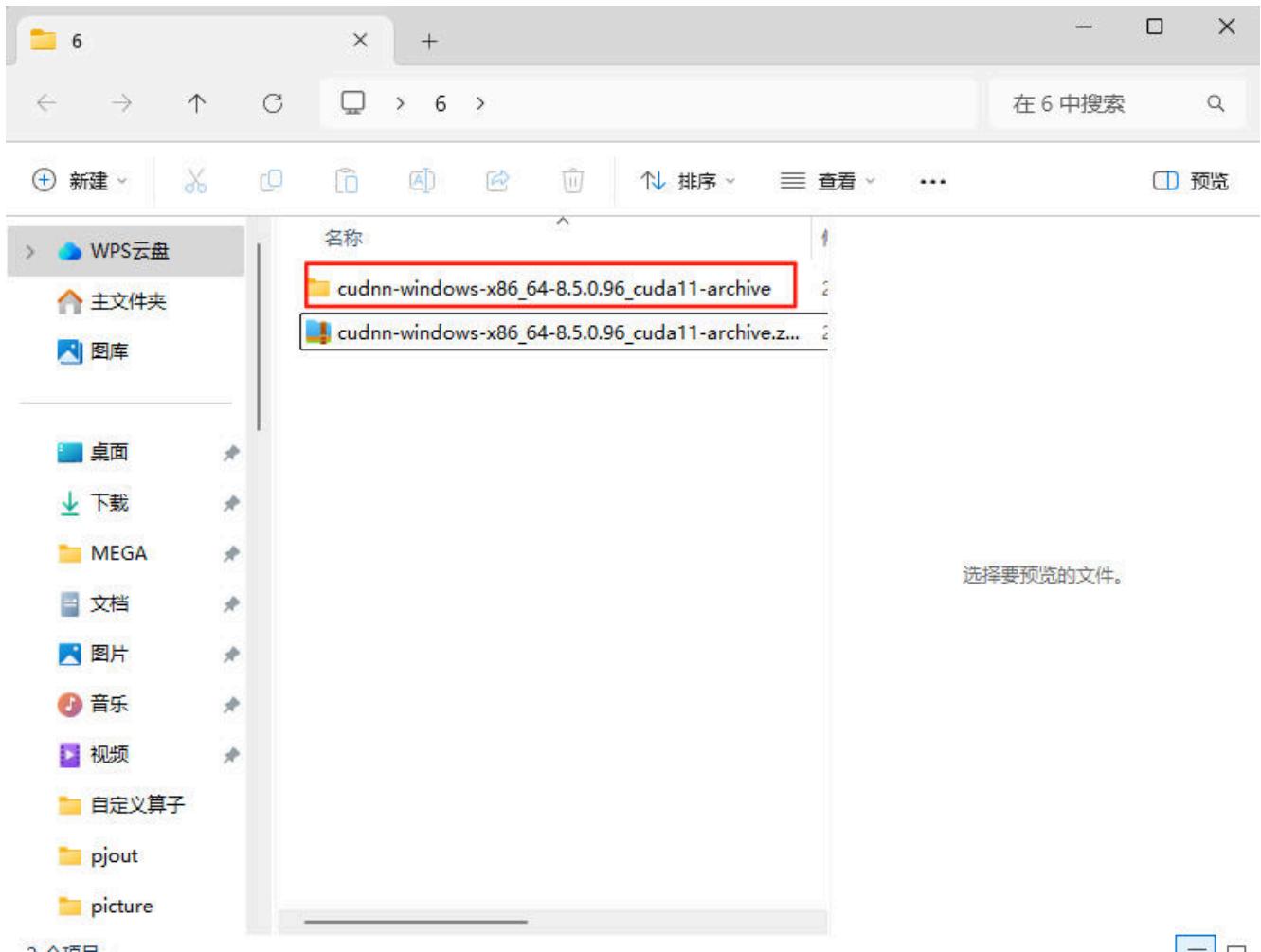
- i. 显卡驱动下载：若电脑在此前从未安装过相应的驱动，需先安装显卡驱动，英伟达显卡驱动[官方下载连接](#)。
 - ii. cuda工具包下载：在安装前现在命令台输入("nvidia-smi.exe")查询**最高支持**的cuda version
(就只是最高而已，不需要一定下载这个，强烈建议就只下载cuda11.7)，如图所示：

```
命令提示符
Microsoft Windows [版本 10.0.22631.5624]
(c) Microsoft Corporation. 保留所有权利。

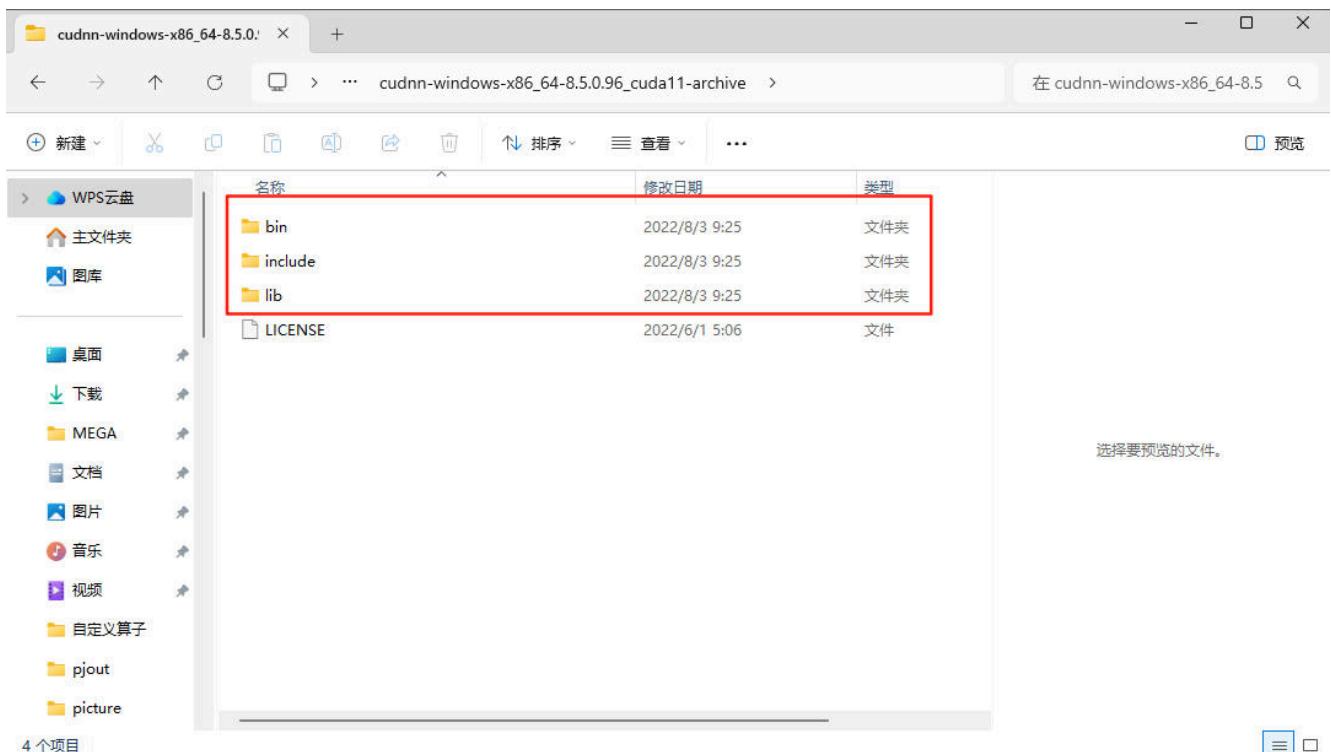
C:\Users\zhuji>nvidia-smi.exe
Wed Jul 16 11:18:01 2025
+
| NVIDIA-SMI 546.30           Driver Version: 546.30      CUDA Version: 12.3 |
| GPU  Name        TCC/WDDM | Bus-Id     Disp.A  Volatile Uncorr. ECC | | | |
| Fan  Temp     Perf  Pwr:Usage/Cap | Memory-Usage | GPU-Util  Compute M. |
|          |          |             |            | GPU-Mem M. |
+-----+-----+-----+-----+-----+-----+-----+-----+
|   0  NVIDIA GeForce RTX 3050 ... WDDM    00000000:01:00.0 On |          N/A | | | |
| N/A  42C     P8          4W /  60W |       640MiB /  4096MiB |      0%     Default |
|          |          |             |            |          N/A |
+-----+-----+-----+-----+-----+-----+-----+-----+
+
| Processes:                               GPU Memory |
| GPU  GI  CI      PID  Type  Process name        Usage  |
| ID   ID              |          |
+-----+-----+-----+-----+-----+-----+-----+
|   0  N/A  N/A      2124  C+G  ... 0_x64_8wekyb3d8bbwe\XboxPCTray.exe  N/A |
|   0  N/A  N/A     10112  C+G  ...1\extracted\runtime\WeChatAppEx.exe  N/A |
|   0  N/A  N/A     10296  C+G  ...on\138.0.3351.83\msedgeWebView2.exe  N/A |
|   0  N/A  N/A     10548  C+G  ...5n1h2txyewy\ShellExperienceHost.exe  N/A |
|   0  N/A  N/A     11176  C+G  C:\Windows\explorer.exe                 N/A |
+
```

在cuda工具包[下载链接](#)下载后安装。(如果是win10的电脑需从官网下载, [下载链接](#))

- iii. cuDNN下载：[下载链接](#)下载完成后解压

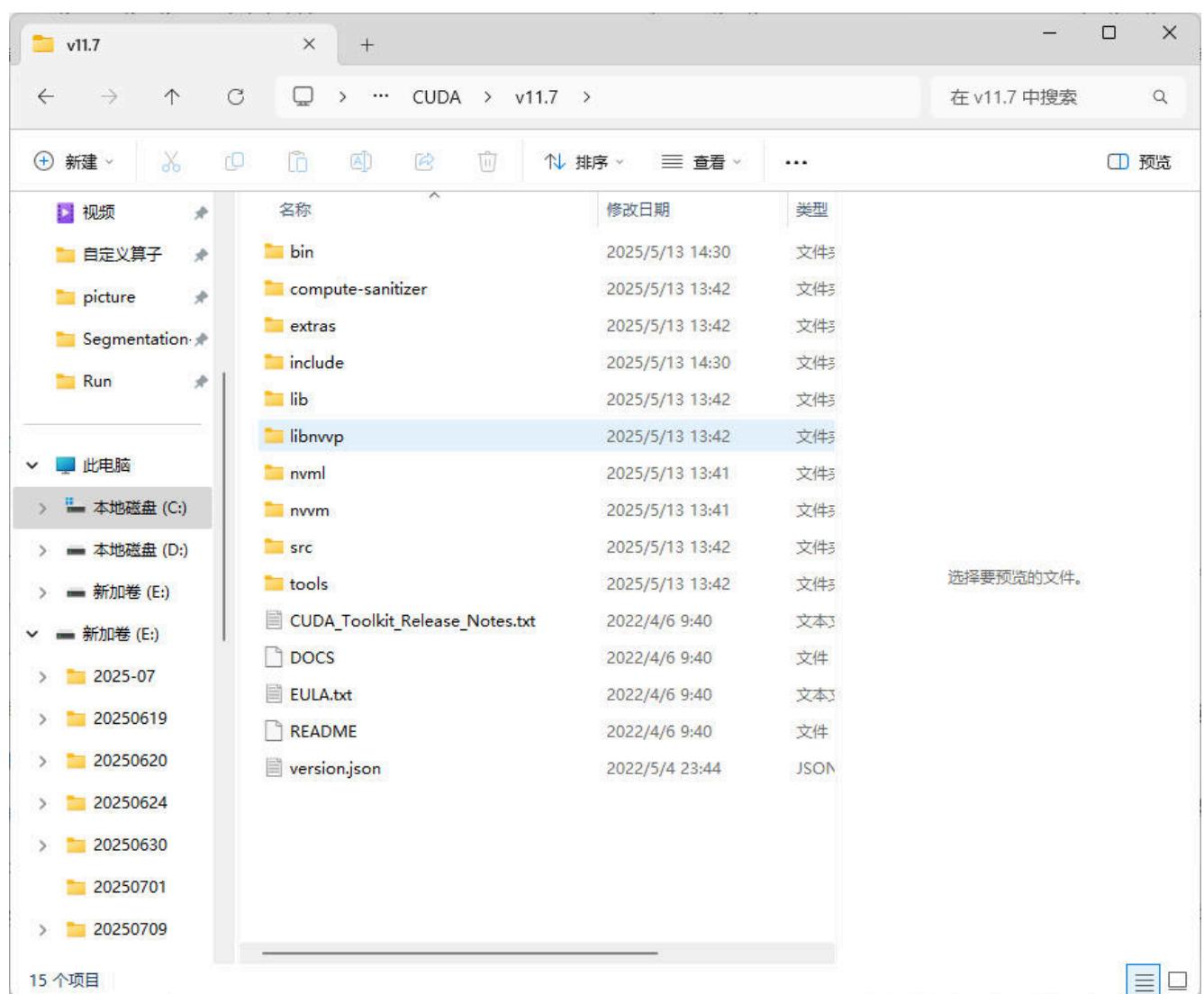
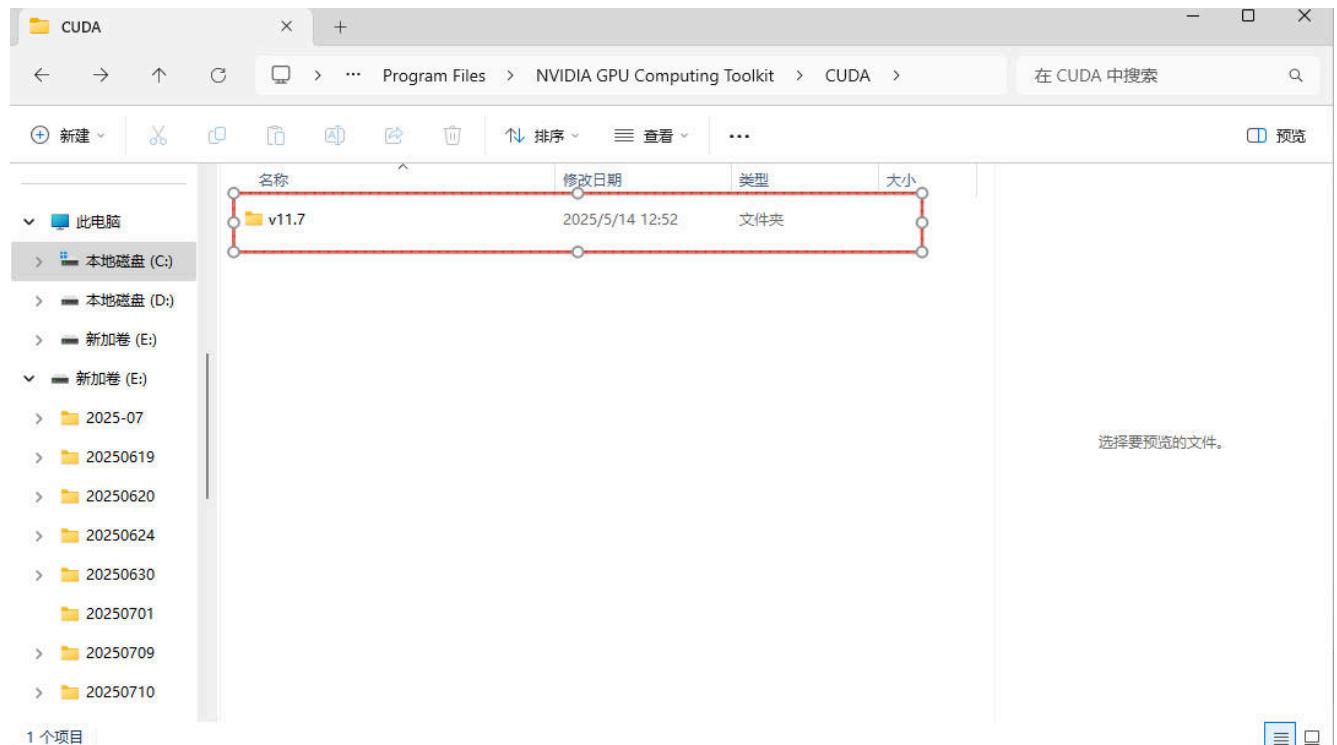


进入解压完的文件夹，复制"bin"、"include"、"lib"这三个文件夹

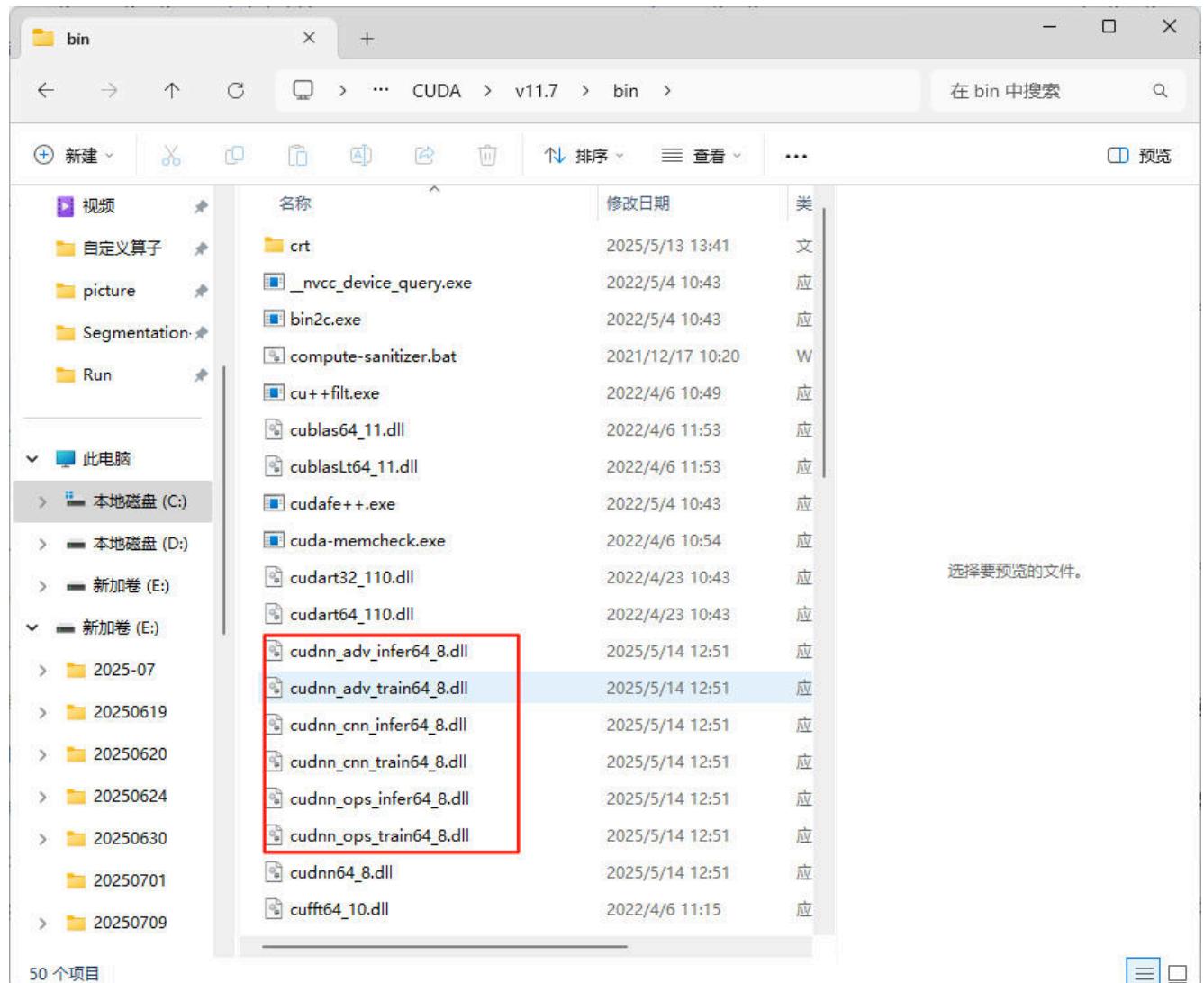


打开“C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA”里面有你安装的对应的cuda工

具包版本，再打开这个文件夹。



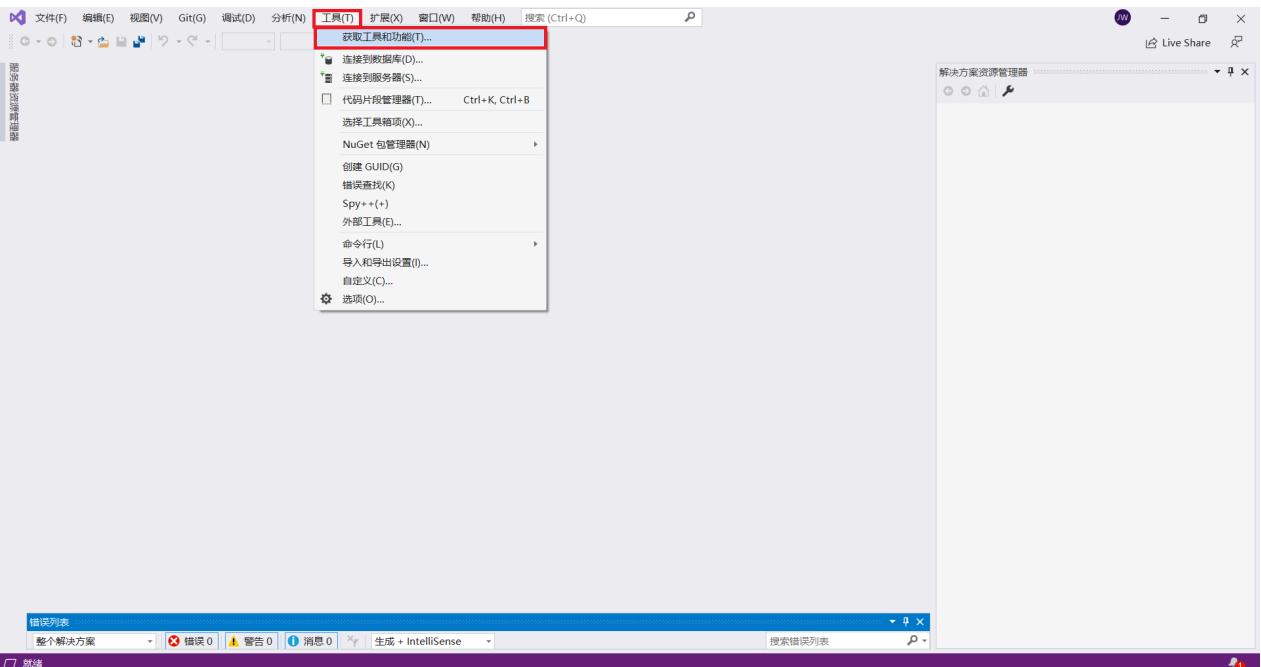
将cuDNN中复制的三个文件夹粘贴进此文件夹。粘贴完后进入"bin"中应该可以看到里面包含 cudnn



2. 环境依赖：Visual Studio 2017版本及以上（建议安装Visual Studio 2022），Visual Studio[下载链接](#)。在官网下载号Visual Studio后进行安装，注意勾选“.Net桌面开发”和“通用Windows平台开发”。

i. 如果您已经下载VS软件：

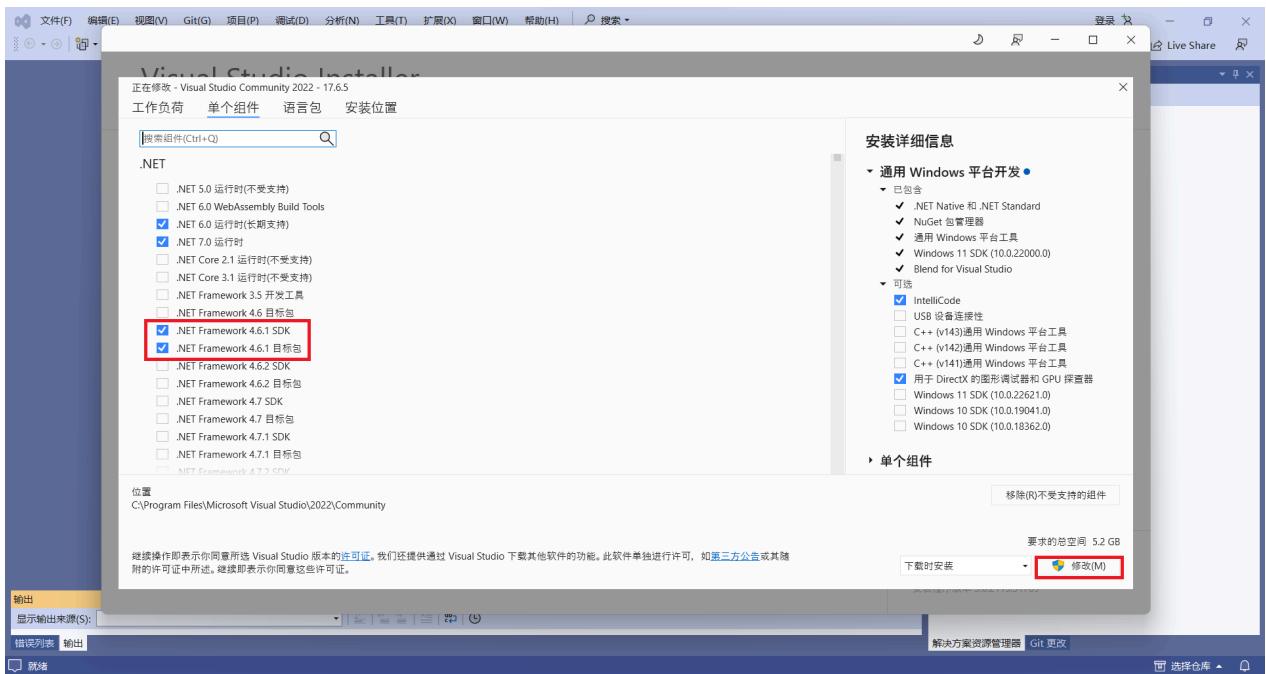
a. 请在“工具”-“获取工具和功能”中下载安装包



b. 在“工作负载”中，勾选“.Net桌面开发”和“通用Windows平台开发”，然后点击“修改”。



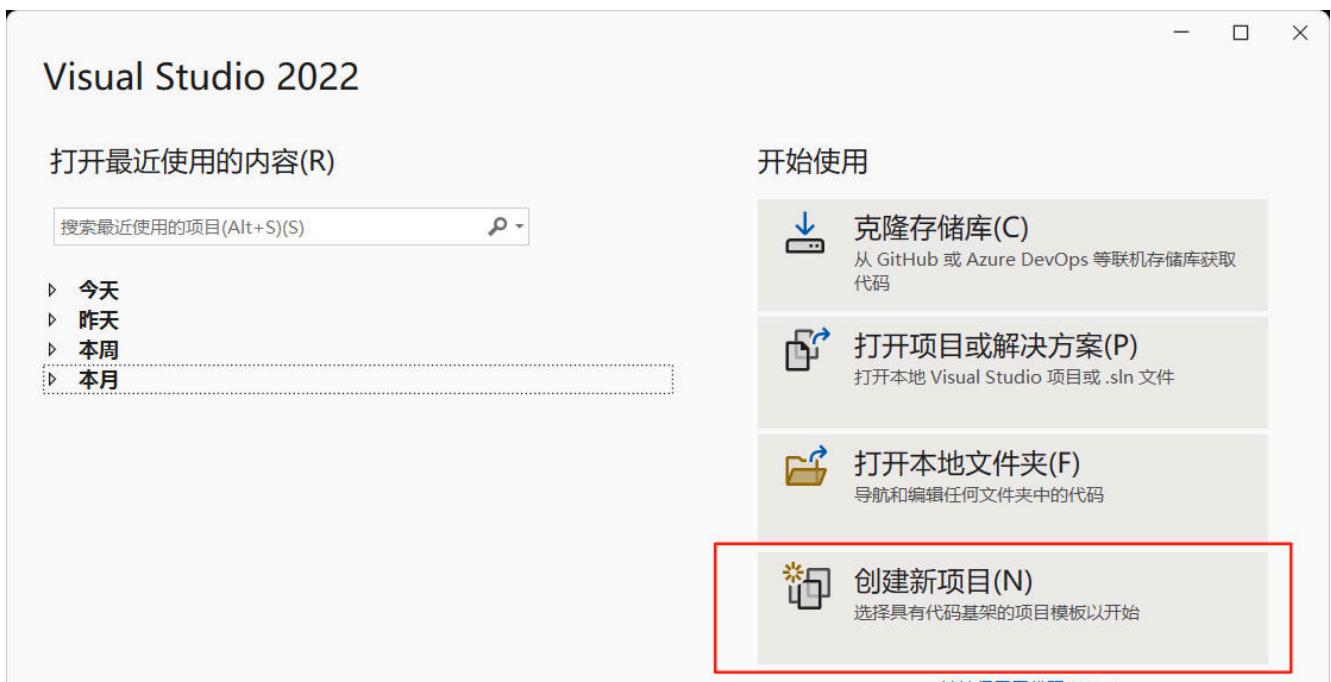
c. 在“单个组件”中，勾选“.Net Framework4.6.1 SDK”和“.NET Framework4.6.1”目标包。然后点击修改



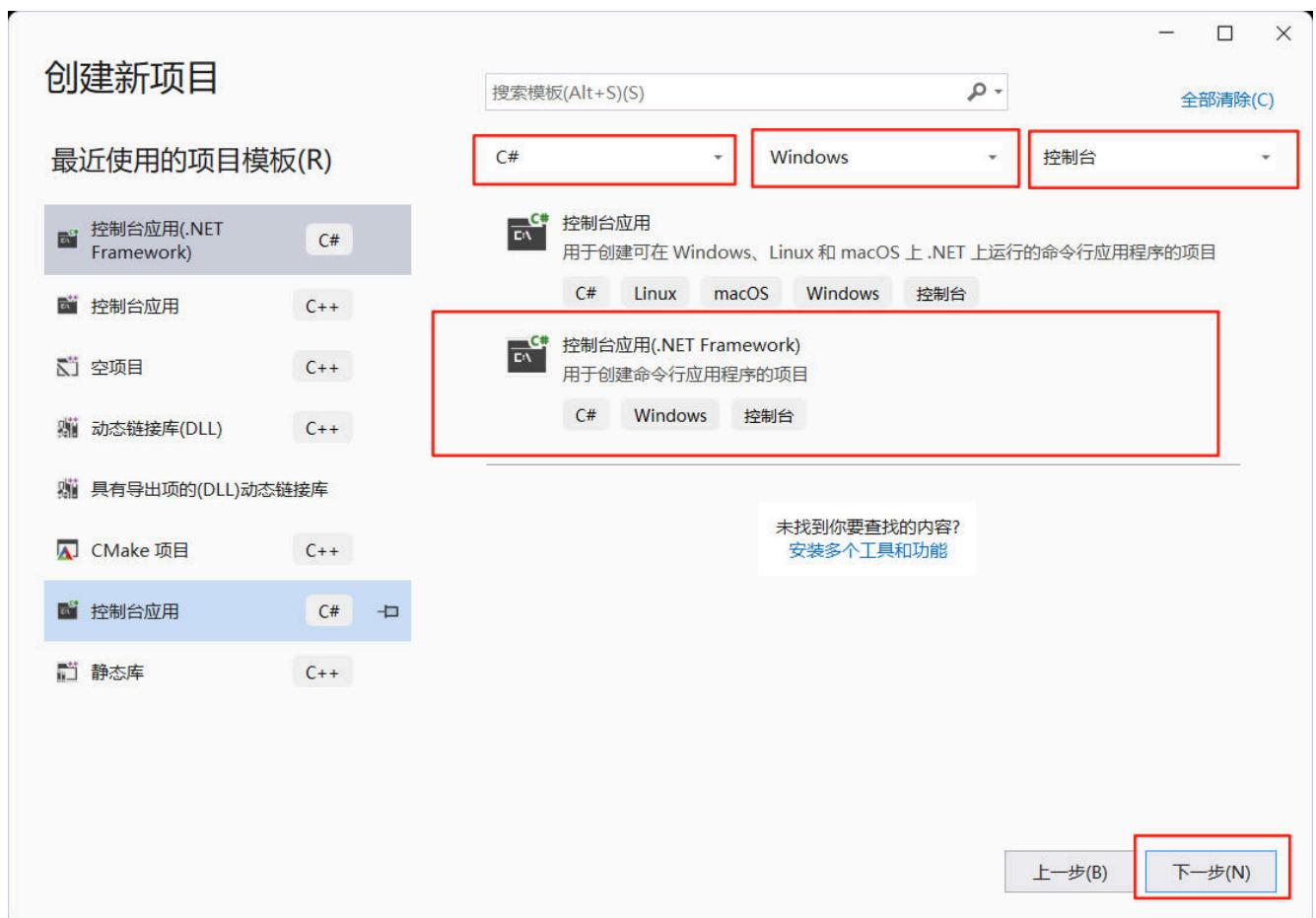
2. 快速使用

1. 新建项目

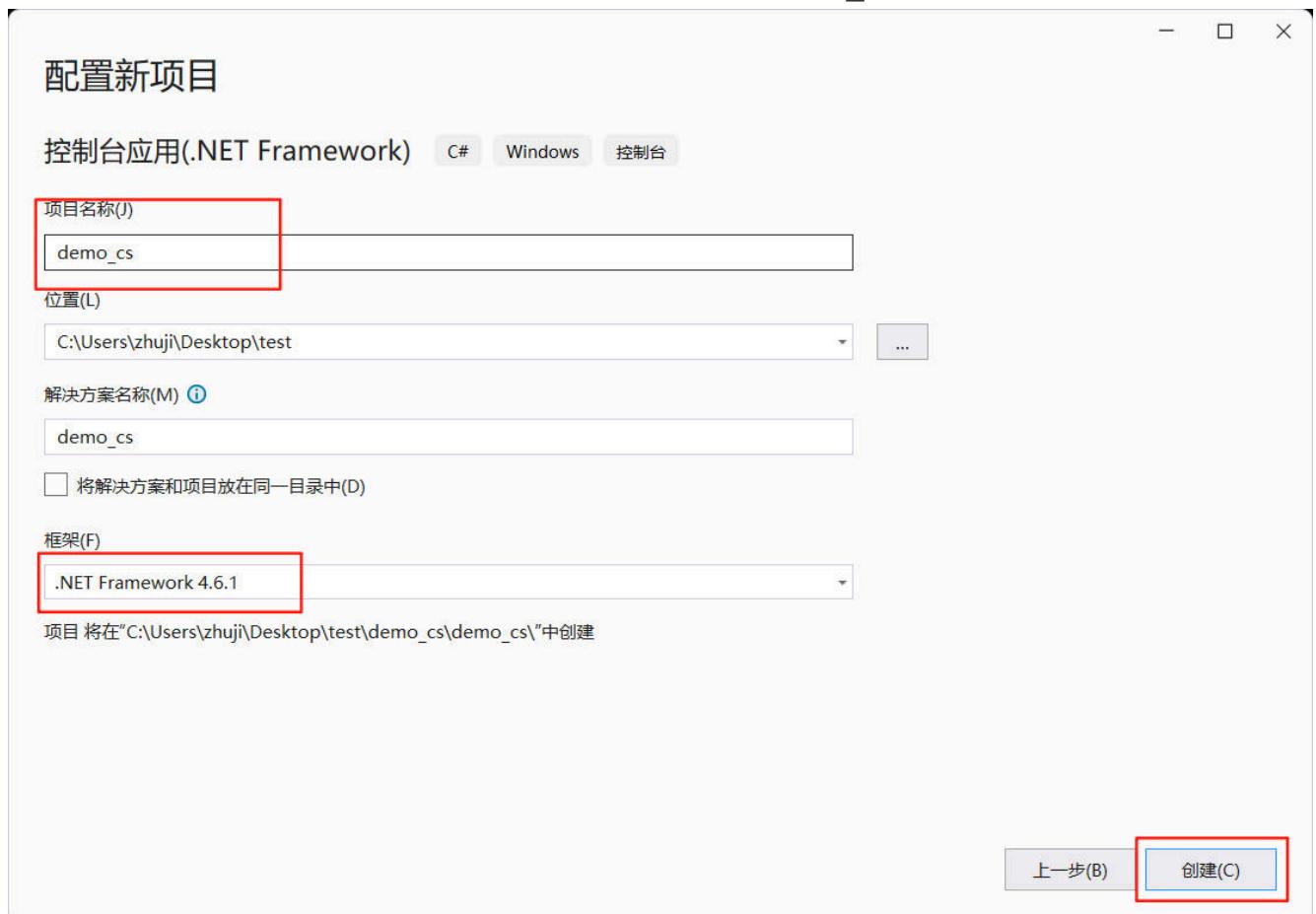
- 打开visual stdio，点击"创建新项目"



- 选择"C#"、"Windows"、"控制台"，然后点击"下一步"

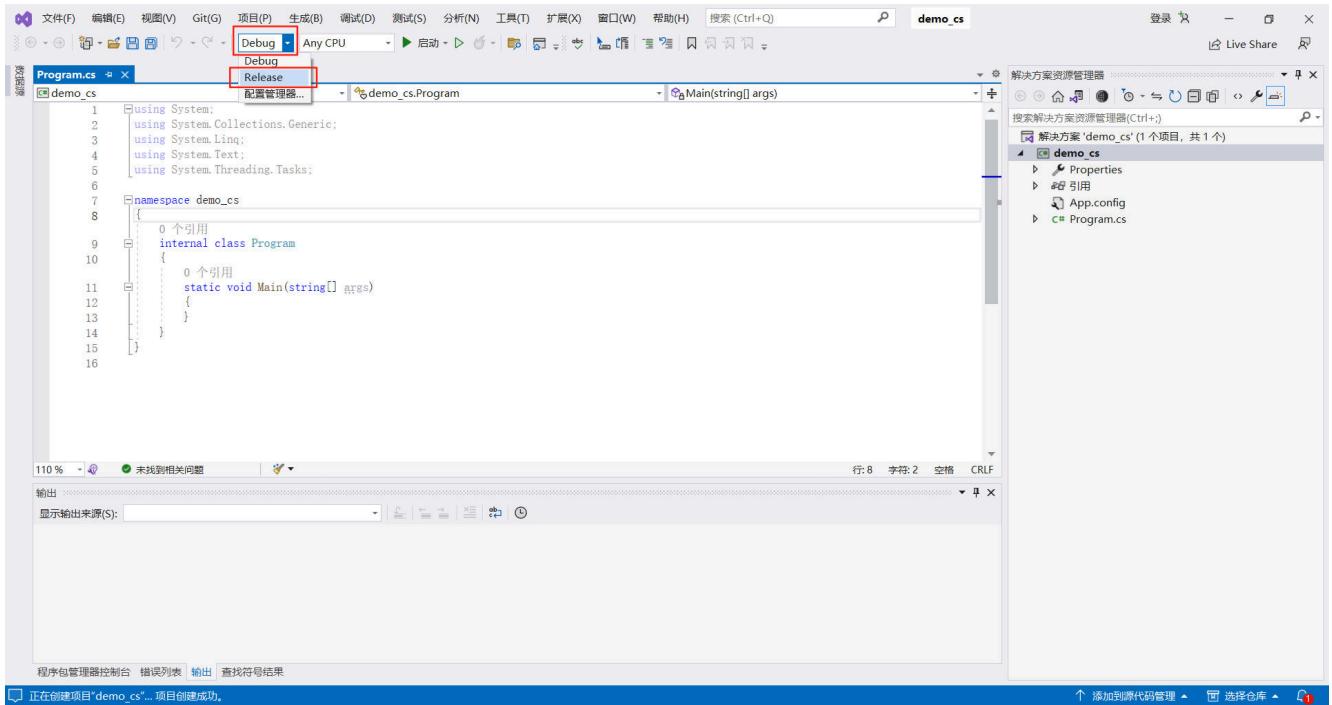


iii. 在“框架”中，选择“.NET Framework 4.6.1”,项目名称为“demo_cs”，点击创建

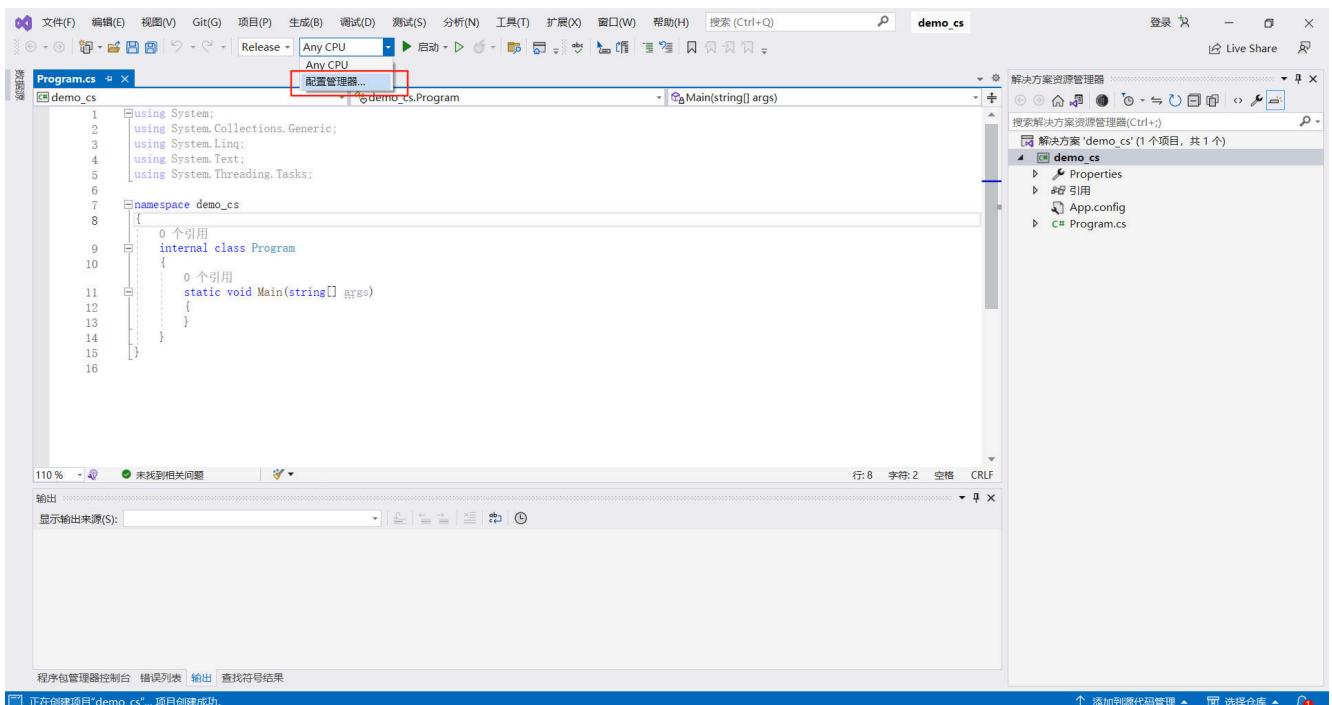


2. 修改运行配置

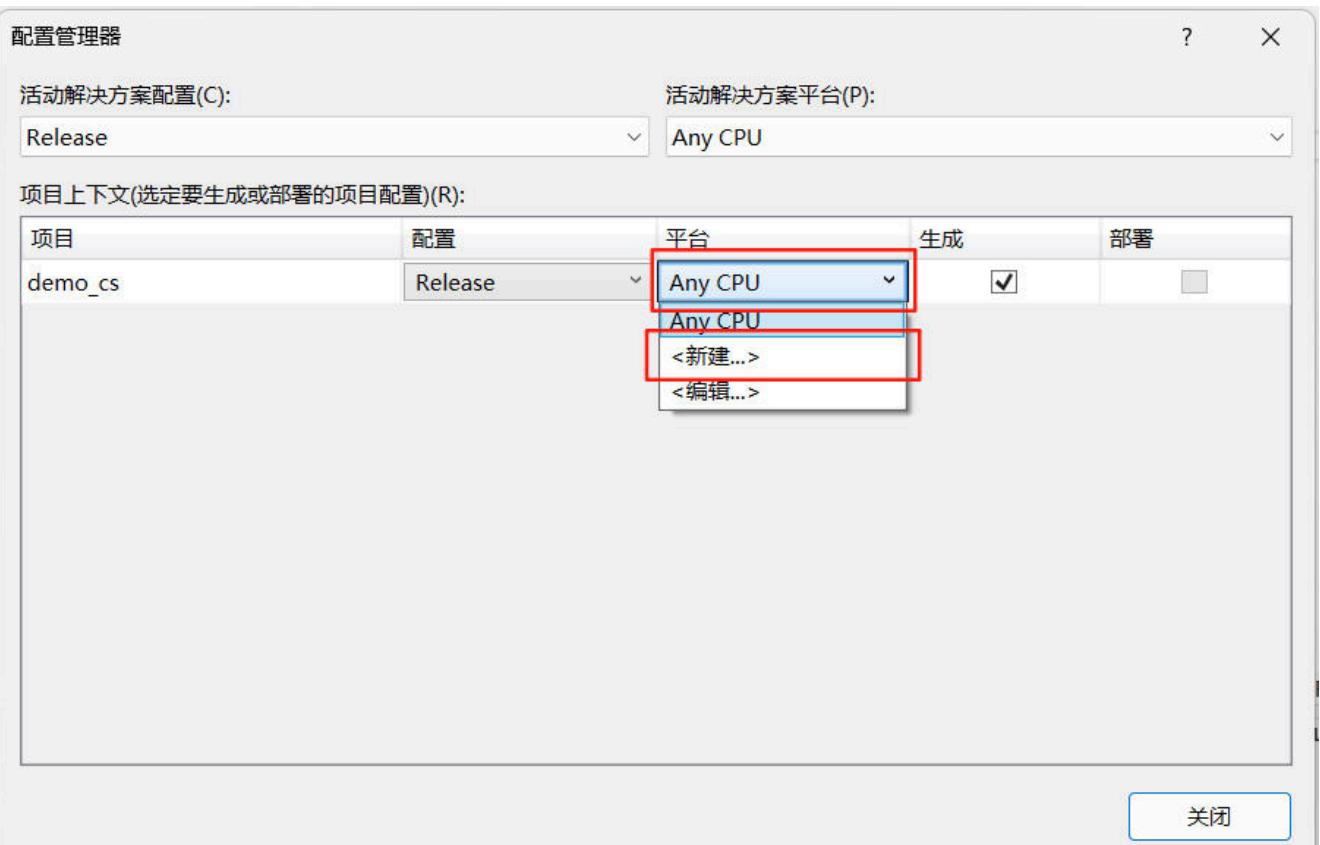
i. 在红框中勾选"Release"



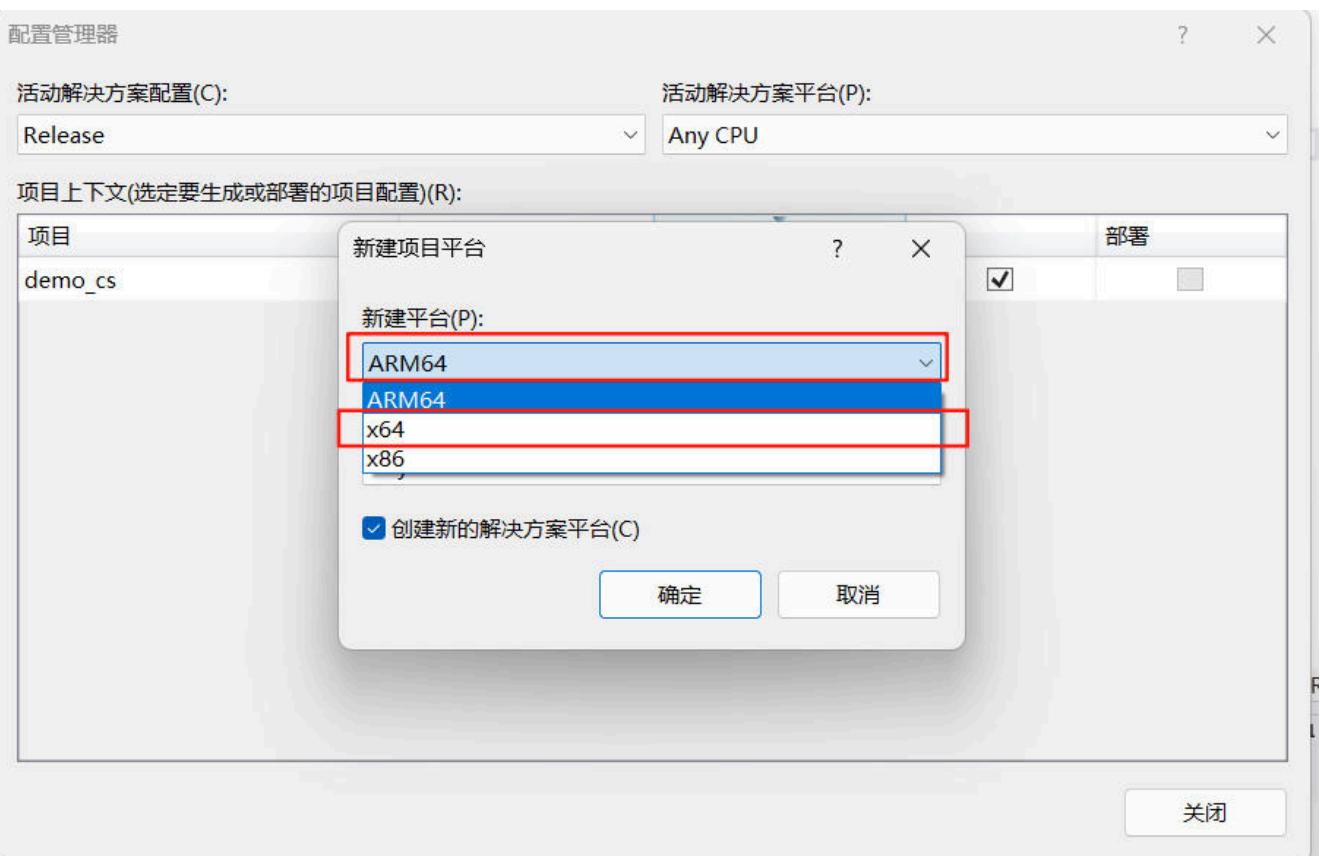
ii. 在"Any CPU"处，点击"配置管理器"



iii. 点击图中红框内下拉菜单，选择"新建"



iv. 选择"x64"

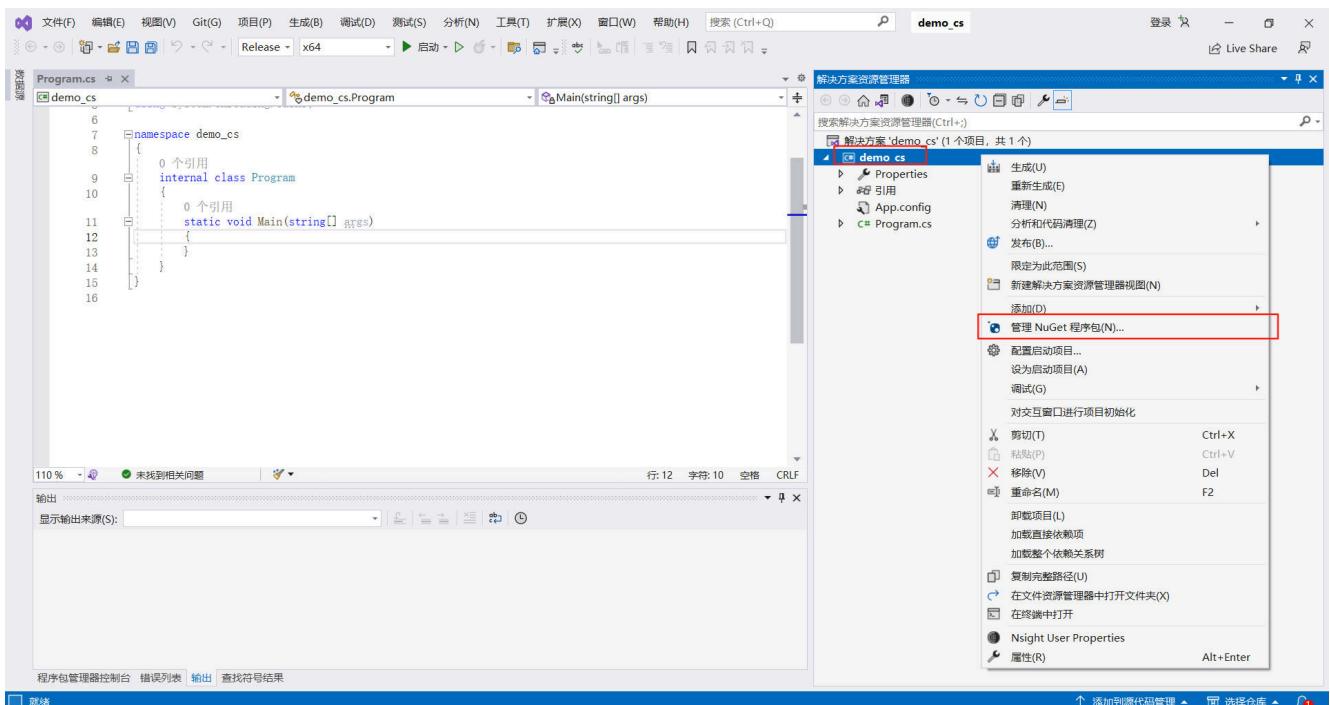


v. 最终效果为

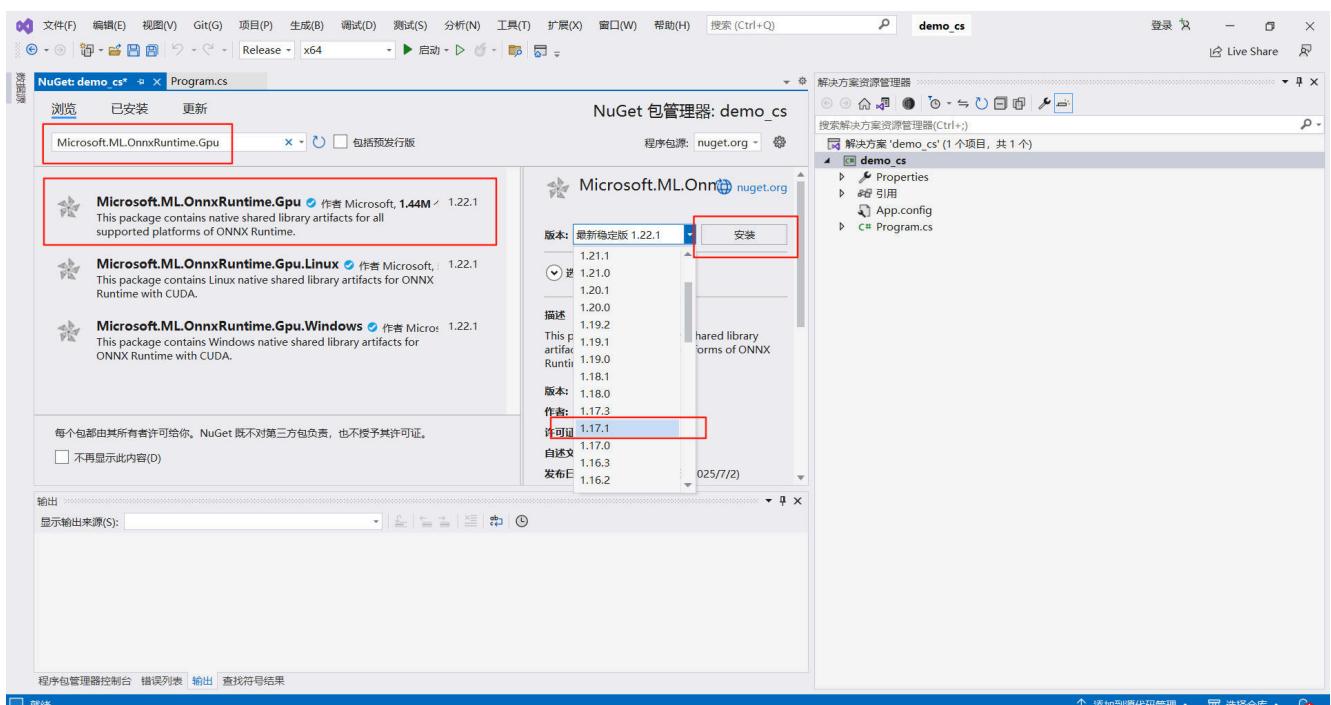
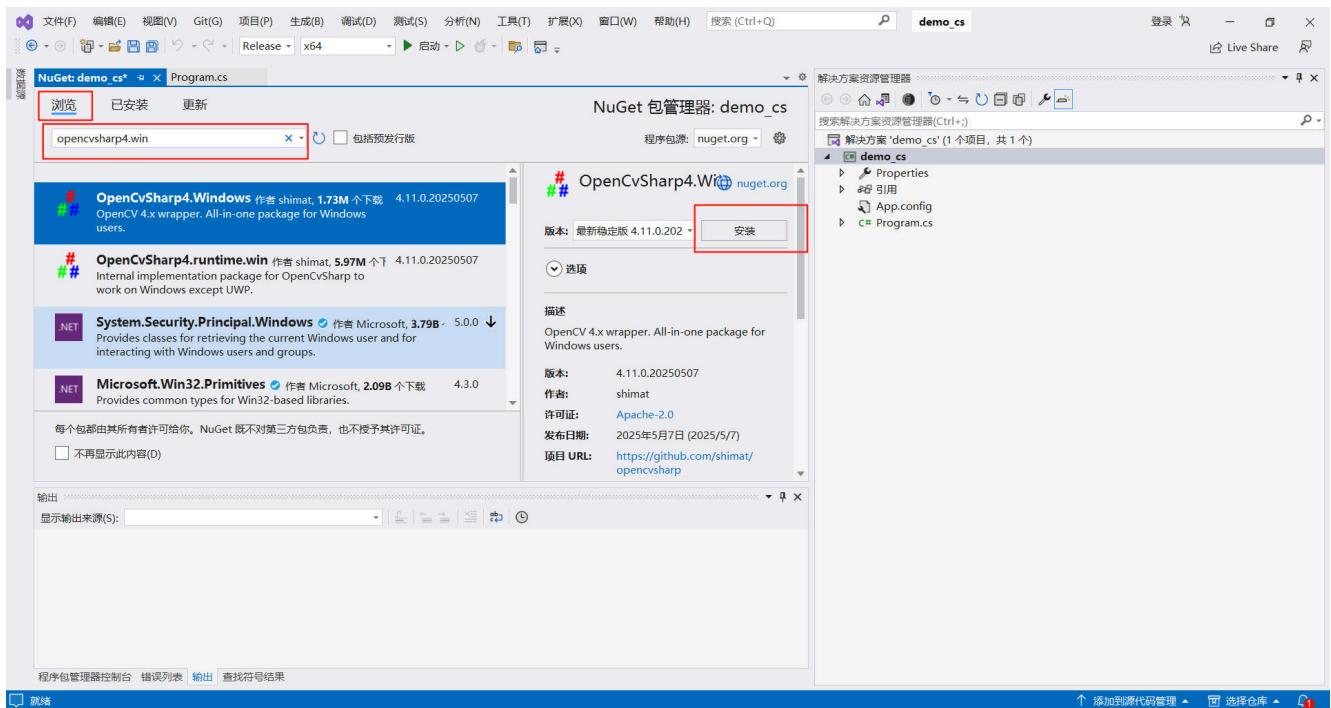
```
1. Using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6.
7. namespace demo_cs
8. {
9.     0 个引用
10.    internal class Program
11.    {
12.        0 个引用
13.        static void Main(string[] args)
14.        {
15.        }
16.    }
}
```

3. 添加NuGet包

- 右键“解决方案资源管理器”中的“demo_cs”，点击“管理NuGet程序包”

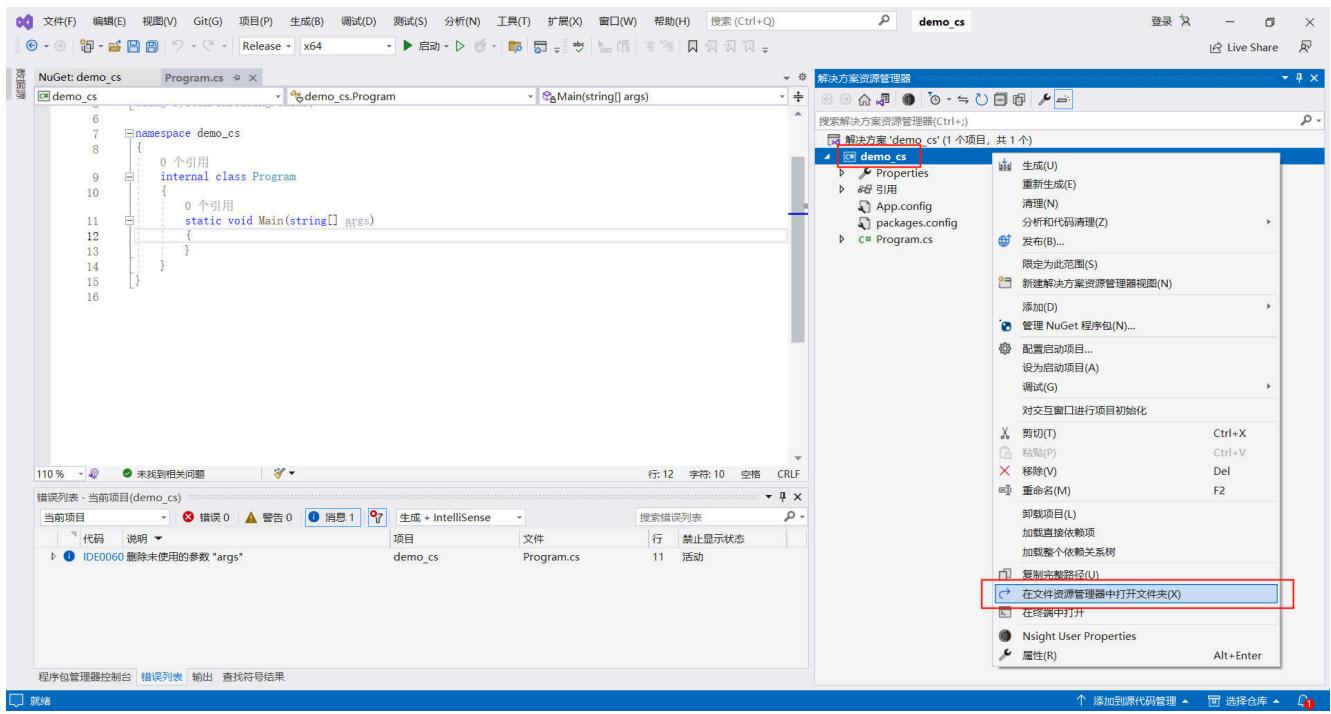


- 添加"openCVsharp4.win"和"Microsoft.ML.OnnxRuntime.Gpu"，其中"Microsoft.ML.OnnxRuntime.Gpu"的版本需要选择1.17.1

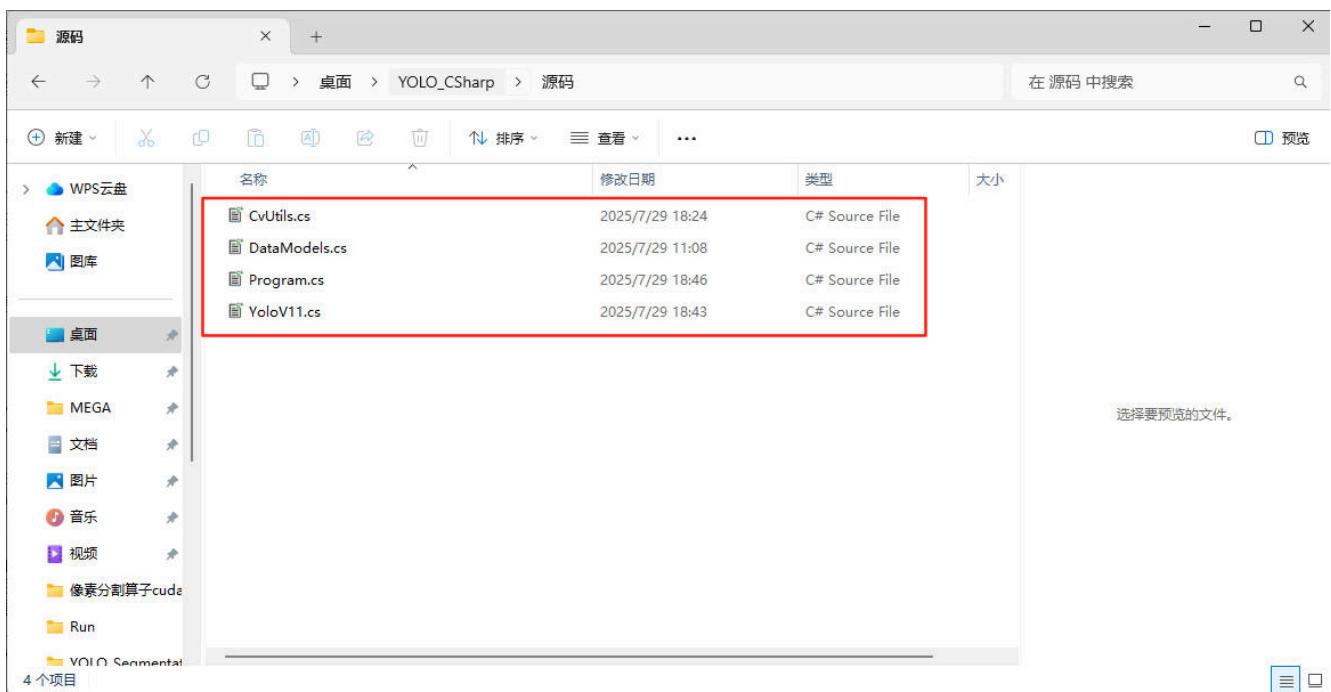


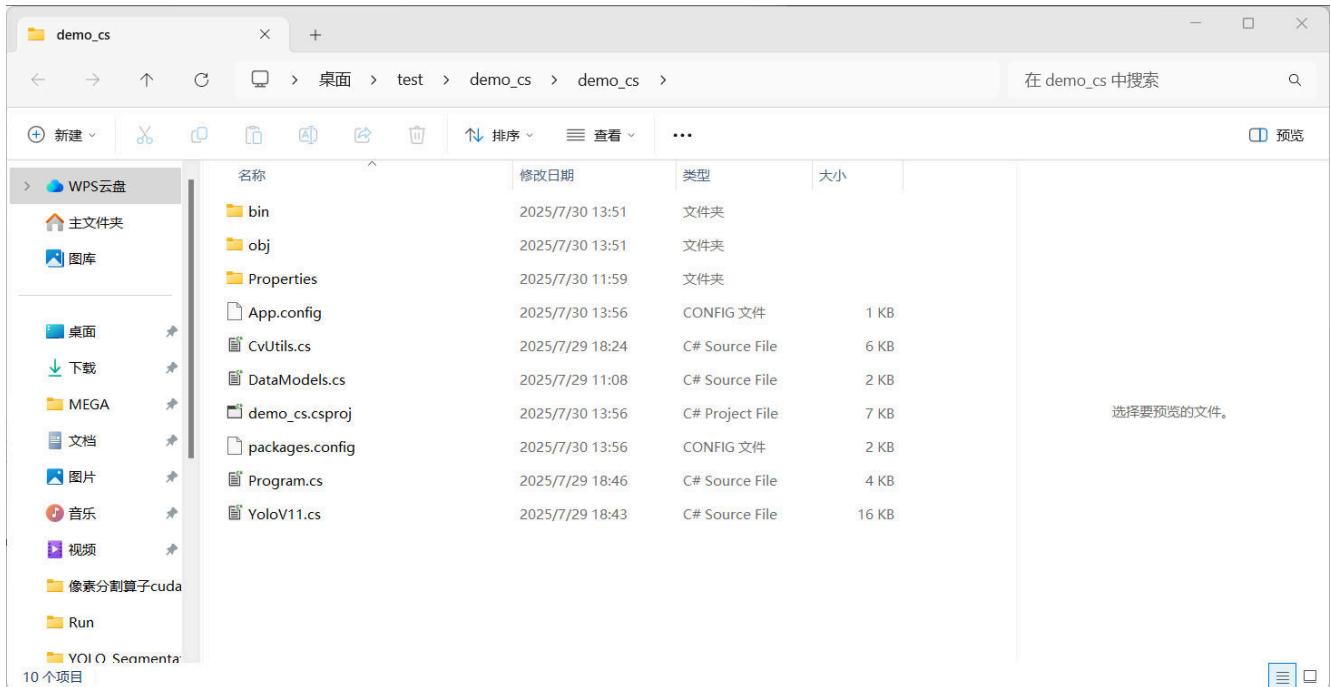
4. 修改代码

- 在"解决资源管理器"中，右键，点击"在文件资源管理器中打开文件夹"。

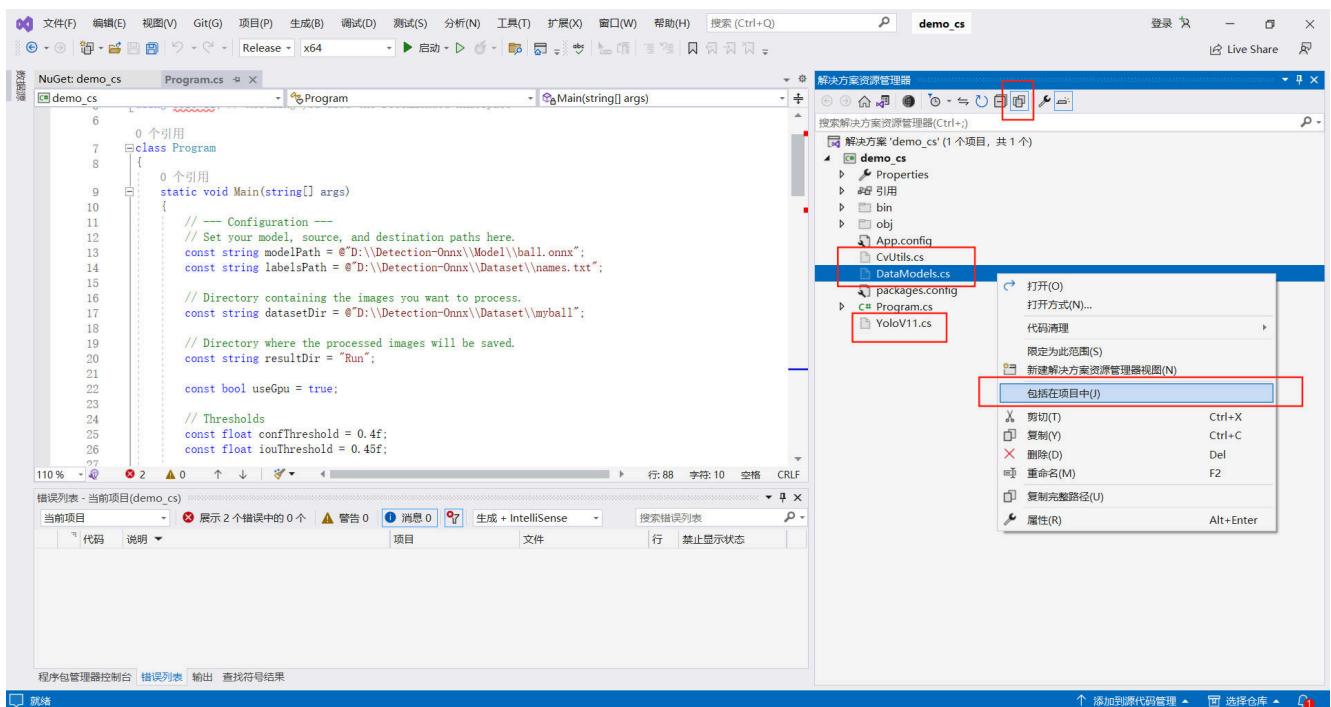


ii. 打开此"readme"下的"code"文件夹，将其中的所有文件复制粘贴到上一步的"在文件资源管理器中打开文件夹"的目录中





iii. 返回Visual Studio中,在"解决方案资源管理器"中,点击"显示所有文件",依次右键"CvUtils.cs"、"DataModels.cs"、"YoloV11.cs"这三个, "包含在项目中"



The screenshot shows the Visual Studio IDE interface. On the left, the code editor displays `CvUtils.cs` with a `Clamp` function implementation. On the right, the Solution Explorer shows the project structure with files like `App.config`, `CvUtils.cs`, `DataModels.cs`, `Program.cs`, and `YoloV11.cs`. The `Program.cs` file is currently selected.

```

using Microsoft.ML.OnnxRuntime.Tensors;
using OpenCvSharp;
using System;
using System.Collections.Generic;
using System.Linq;
// 引入 Marshal 类所在的命名空间
using System.Runtime.InteropServices;
using YoloV11.Models;

namespace YoloV11.Utils
{
    public static class CvUtils
    {
        public static T Clamp<T>(T val, T low, T high) where T : IComparable<T>
        {
            if (val.CompareTo(low) < 0) return low;
            if (val.CompareTo(high) > 0) return high;
            return val;
        }

        public static Mat LetterBox(Mat image, Size newSize, out Size newUnpad, out float ratio)
    }
}

```

iv. "修改Program.cs"中的modelPath、labelsPath、datasetDir、resultDir、useGpu

- modelPath是模型文件的路径
- labelsPath是标签路径（即names.txt）的路径
- datasetDir是待检测图片的路径
- resultDir是检测完图片保存图片的路径
- useGpu是否使用cuda进行推测（如果最开始没有安装cuda工具包和cudnn只能选false）

The screenshot shows the Visual Studio IDE interface. The code editor displays `Program.cs` with a `Main` method. A red box highlights the configuration parameters: `modelPath`, `labelsPath`, `datasetDir`, and `resultDir`. Another red box highlights the `useGpu` variable. The Solution Explorer on the right shows the project structure.

```

using System.IO; // Required for file and directory operations
using System.Linq; // Required for filtering file extensions
using YoloV11; // Assuming you used the recommended namespace

class Program
{
    static void Main(string[] args)
    {
        // Configuration
        const string modelPath = @"D:\Detection-Onnx\Model\ball.onnx";
        const string labelsPath = @"D:\Detection-Onnx\Dataset\names.txt";

        // Directory containing the images you want to process.
        const string datasetDir = @"D:\Detection-Onnx\Dataset\myball";

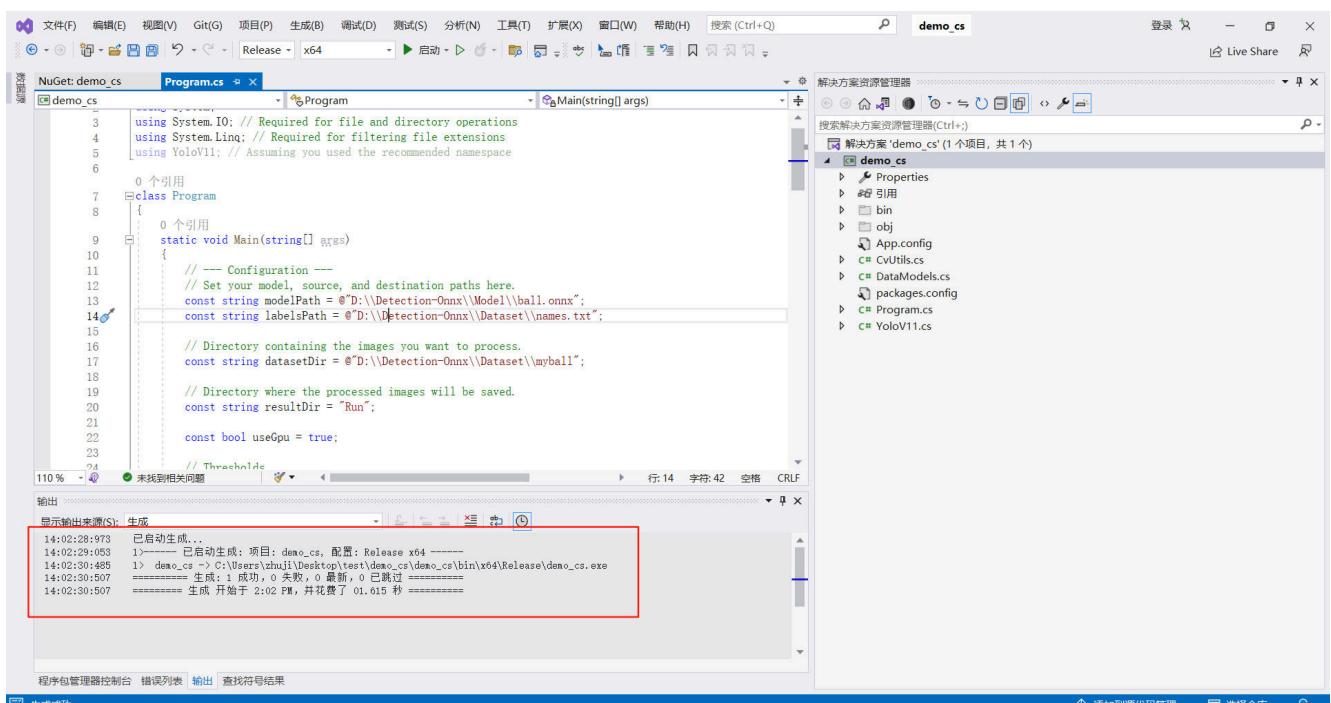
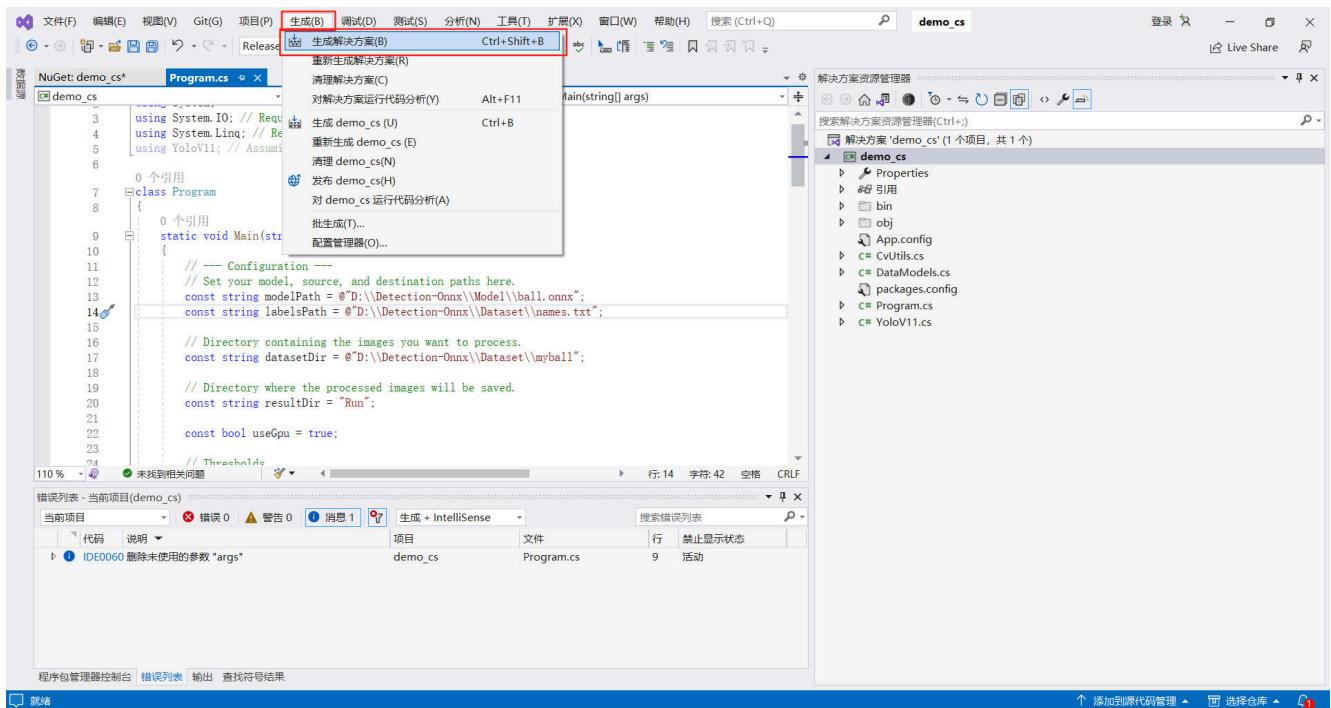
        // Directory where the processed images will be saved.
        const string resultDir = "Run";

        const bool useGpu = true;
    }
}

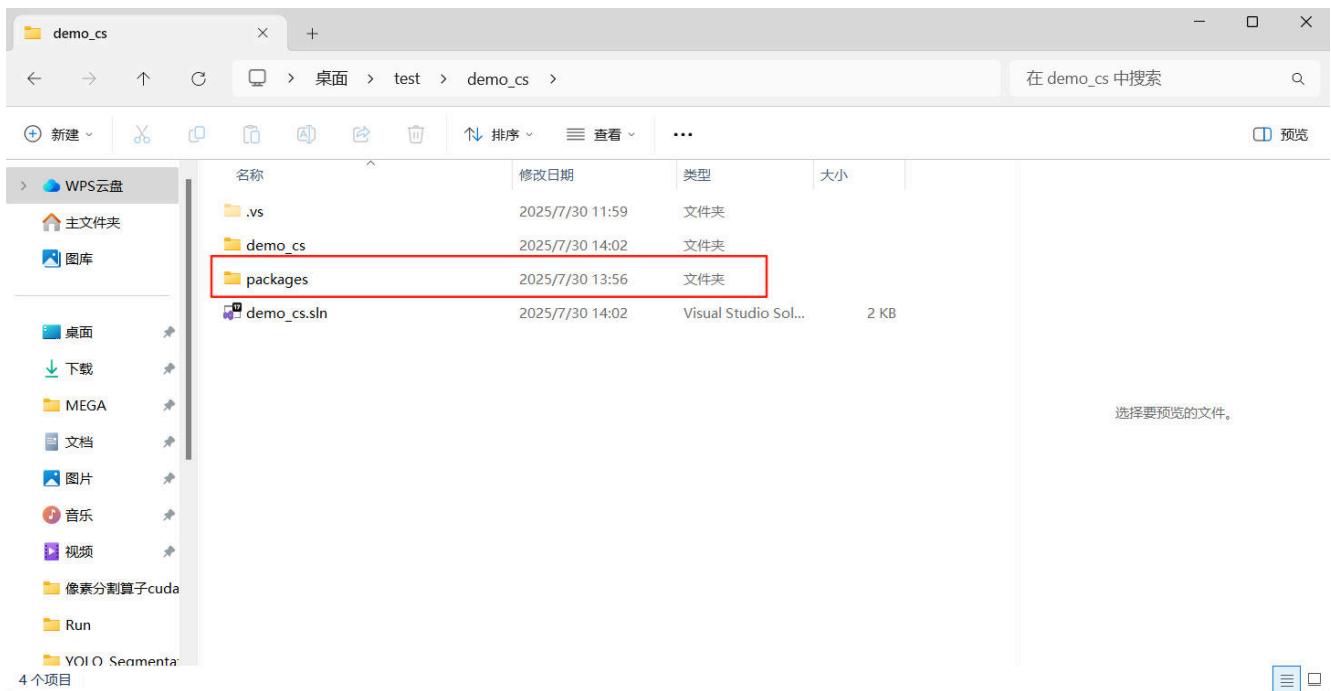
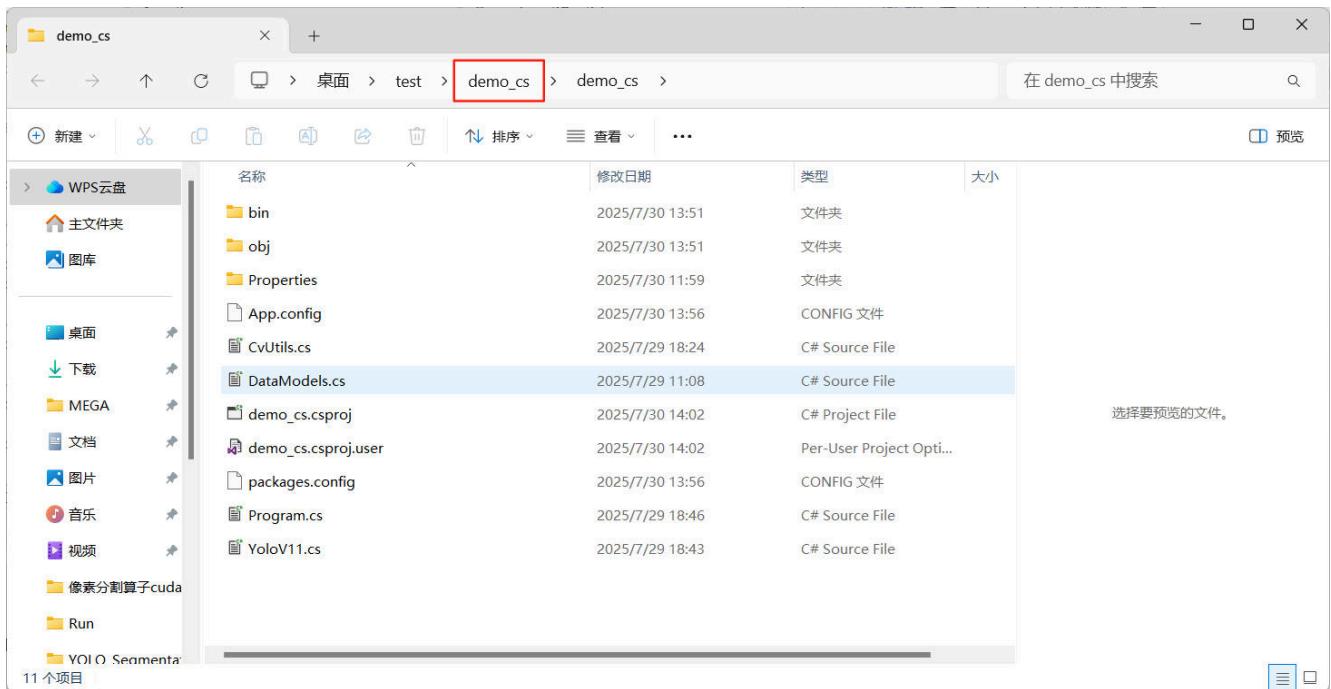
```

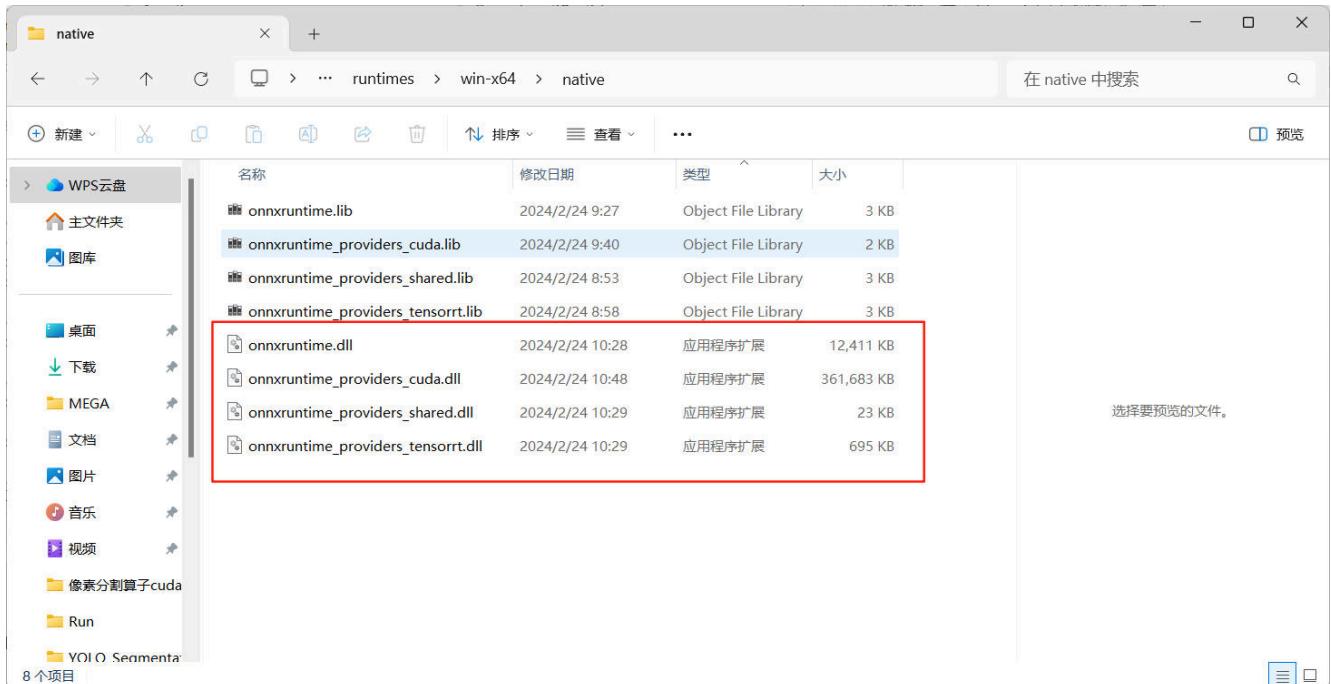
5. 编译

- 点击"生成"-生成解决方案

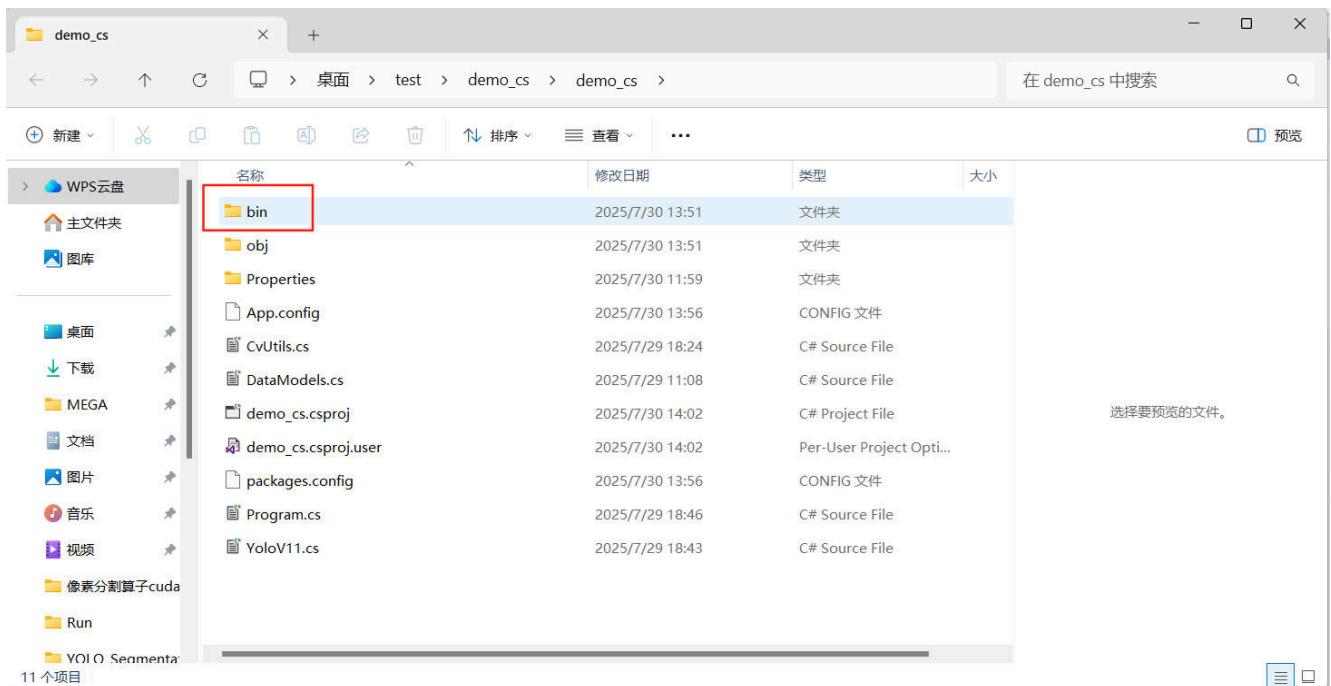


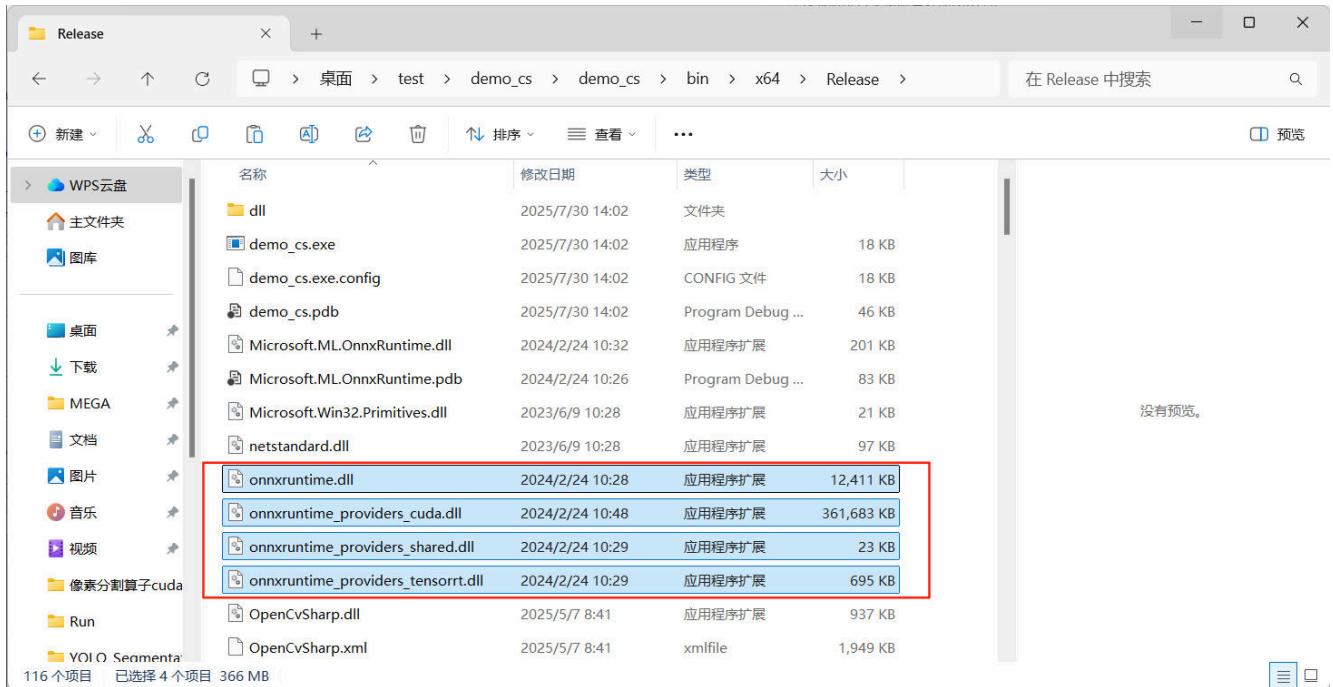
- ii. 在"解决资源管理器"中，右键，点击"在文件资源管理器中打开文件夹"，点击上一级，点击"packages"->"Microsoft.ML.OnnxRuntime.Gpu.Windows.1.17.1"->"runtimes"->"win-x64"->"native"，复制其中所有的.dll文件



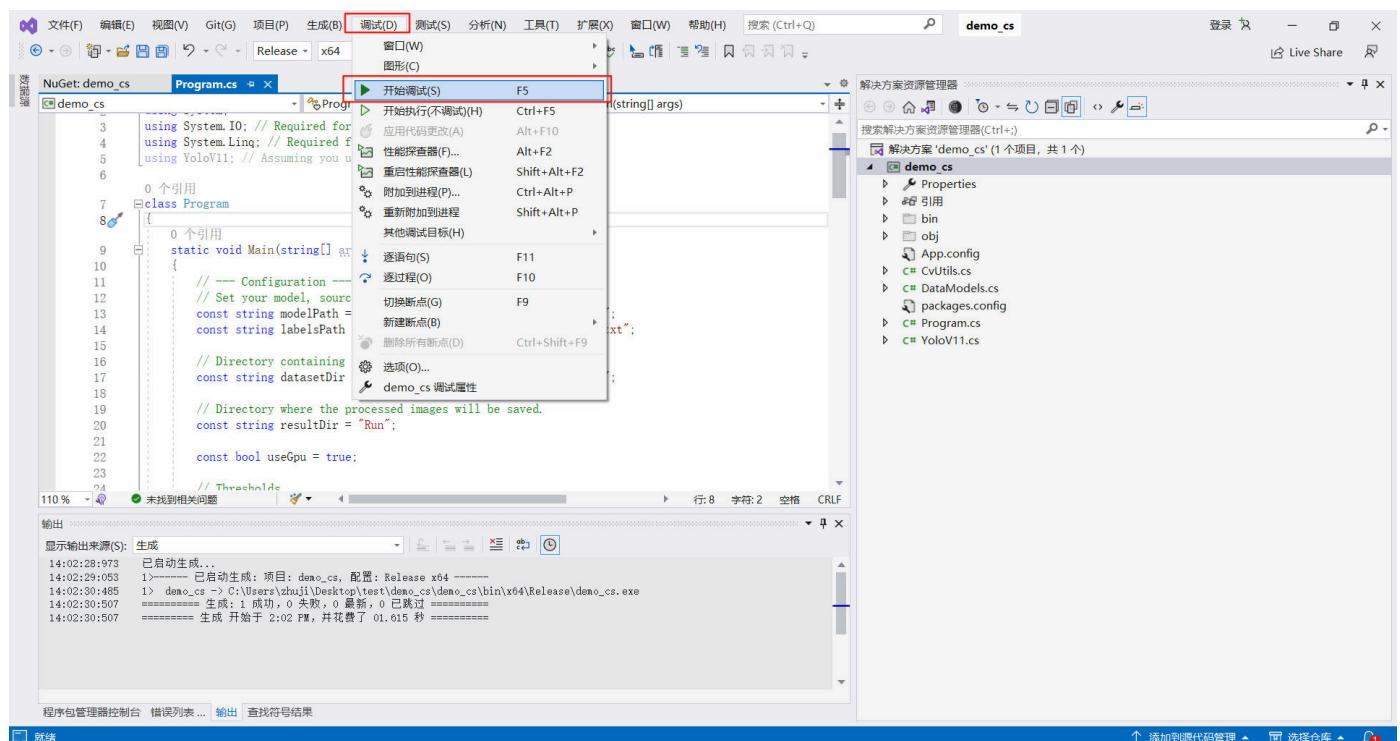


iii. 返回Visual Studio，在“解决资源管理器”中，右键，点击“在文件资源管理器中打开文件夹”，打开“bin”-“x64”-“Release”，粘贴进此文件夹中。





6. 返回Visual Studio中，点击“调试”-“开始执行不调试”



```
C:\Windows\system32\cmd.exe + -
```

[INFO] Using GPU (CUDA) for YOLOv11 inference.
2025-07-30 14:06:55.0948045 [W:onnxruntime:, transformerMemcpy.cc:74 onnxruntime::MemcpyTransformer::ApplyImpl] 4 Memcp
y nodes are added to the graph main_graph for CUDAExecutionProvider. It might have negative impact on performance (inclu
ding unable to run CUDA graph). Set session_options.log_severity_level=1 to see the detail logs before this message.
2025-07-30 14:06:55.1055144 [W:onnxruntime:, session_state.cc:1166 onnxruntime::VerifyEachNodeIsAssignedToAnEp] Some nod
es were not assigned to the preferred execution providers which may or may not have an negative impact on performance. e
.g. ORT explicitly assigns shape related ops to CPU to improve perf.
2025-07-30 14:06:55.1113267 [W:onnxruntime:, session_state.cc:1168 onnxruntime::VerifyEachNodeIsAssignedToAnEp] Rerunnin
g with verbose output on a non-minimal build will show node assignments.
[INFO] YOLOv11 loaded: ball.onnx in 1535 ms
Model Type: Detection
Input shape: Size { Width = 1216, Height = 192 }
#Classes: 9
Found 12 images to process...
Processed: 0A014545A4590D9CDE37E2C9403A6545720054.bmp, saved to Run\0A014545A4590D9CDE37E2C9403A6545720054.bmp
Processed: 0BC3C713842A618AD99B1051AD44EDAA720054.bmp, saved to Run\0BC3C713842A618AD99B1051AD44EDAA720054.bmp
Processed: 0F07E4DEA7E80FE82E6A4BB342B5488F720054.bmp, saved to Run\0F07E4DEA7E80FE82E6A4BB342B5488F720054.bmp
Processed: 0F1B055889B6359A36EFEF0F27A20BF0720054.bmp, saved to Run\0F1B055889B6359A36EFEF0F27A20BF0720054.bmp
Processed: 0FEF4801DE0D035D3518F8BAD8391FBC720054.bmp, saved to Run\0FEF4801DE0D035D3518F8BAD8391FBC720054.bmp
Processed: 1AB647BE1FE0202B4F3F65097020841A720054.bmp, saved to Run\1AB647BE1FE0202B4F3F65097020841A720054.bmp
Processed: 1AC28303C907489EBB290F6DBB643E6F720054.bmp, saved to Run\1AC28303C907489EBB290F6DBB643E6F720054.bmp
Processed: 1B074ED9B09B7FA9A40054CEE8A067D1720054.bmp, saved to Run\1B074ED9B09B7FA9A40054CEE8A067D1720054.bmp
Processed: 1BC62F00BF2064468B886F5B5FD02C1B720054.bmp, saved to Run\1BC62F00BF2064468B886F5B5FD02C1B720054.bmp
Processed: 1D8DF5BBD787F8EE2AD849C6A1708872720054.bmp, saved to Run\1D8DF5BBD787F8EE2AD849C6A1708872720054.bmp
Processed: 1DE12912E96C7AB3B6EBBD4F685BEE20720054.bmp, saved to Run\1DE12912E96C7AB3B6EBBD4F685BEE20720054.bmp
Processed: 1E291CA432CEC84F64DDAF144608E4C2720054.bmp, saved to Run\1E291CA432CEC84F64DDAF144608E4C2720054.bmp

Processing complete.
请按任意键继续... |