

基于yolo的像素分割（海康算子输出至Blob标签分析）使用文档

2. 环境配置

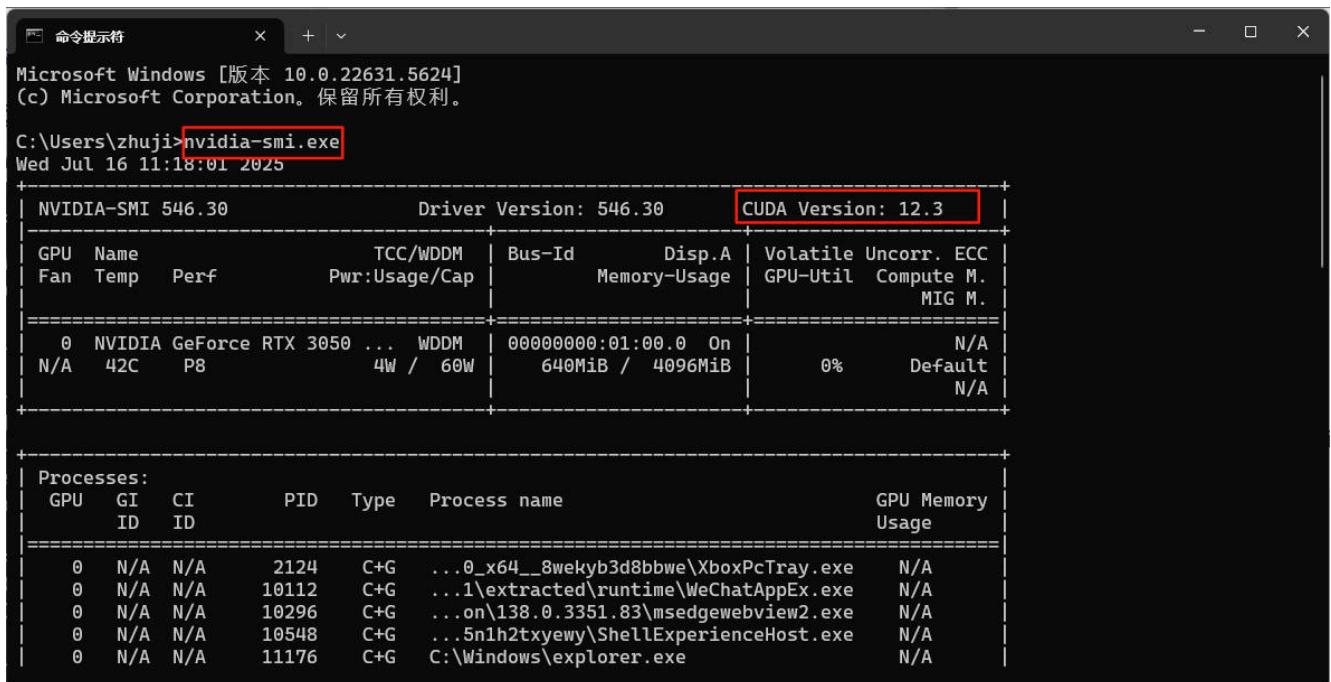
- 支持的操作系统：Windows10/11 X64

此项目只支持在Nvidia显卡上运行。

- 运行时依赖：此项目需要使用cuda,因此需要安装cuda toolkit以及cuDNN。

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[下载连接](#)。

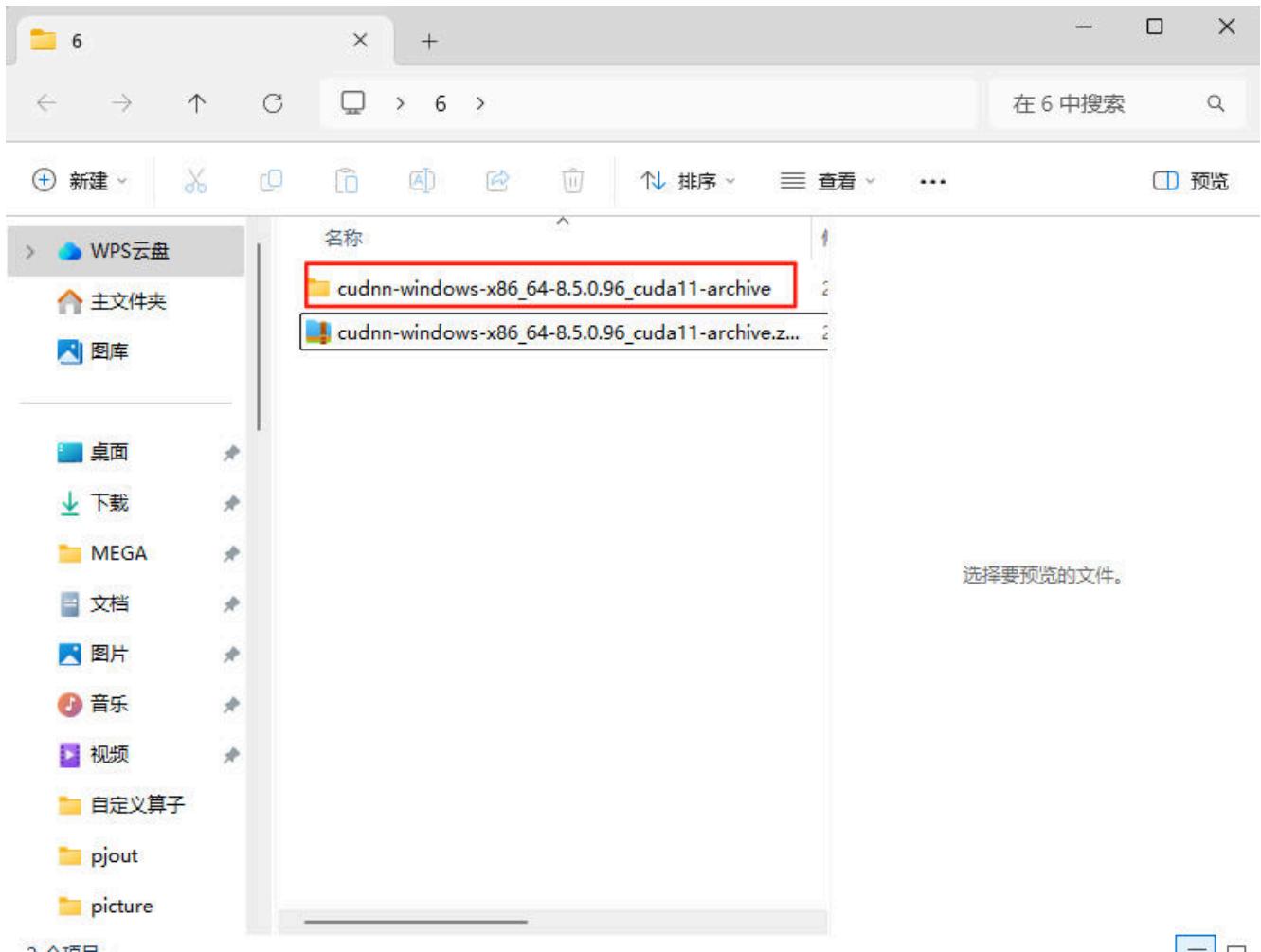
ii. cuda工具包下载：在安装前现在命令台输入("nvidia-smi.exe")查询**最高支持**的cuda version (就只是最高而已，不需要一定下载这个，强烈建议就只下载cuda11.7) , 如图所示：



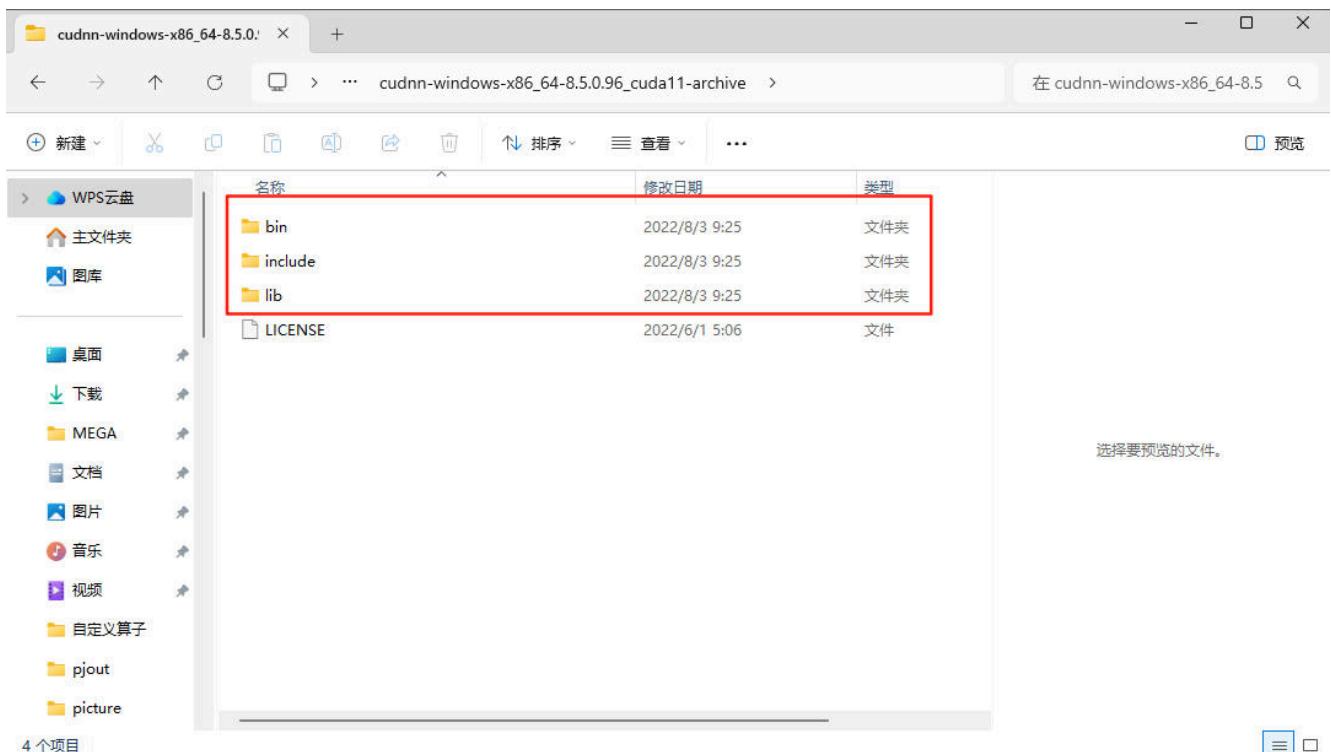
```
命令提示符
Microsoft Windows [版本 10.0.22631.5624]
(c) Microsoft Corporation。保留所有权利。
C:\Users\zhuji>nvidia-smi.exe
Wed Jul 16 11:18:01 2025
+-----+
| NVIDIA-SMI 546.30       Driver Version: 546.30     CUDA Version: 12.3 |
+-----+
| GPU  Name        TCC/WDDM   Bus-Id Disp.A  Volatile Uncorr. ECC |
| Fan  Temp   Perf Pwr:Usage/Cap | Memory-Usage | GPU-Util  Compute M. |
|          |          |          |          |          |          |          MIG 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

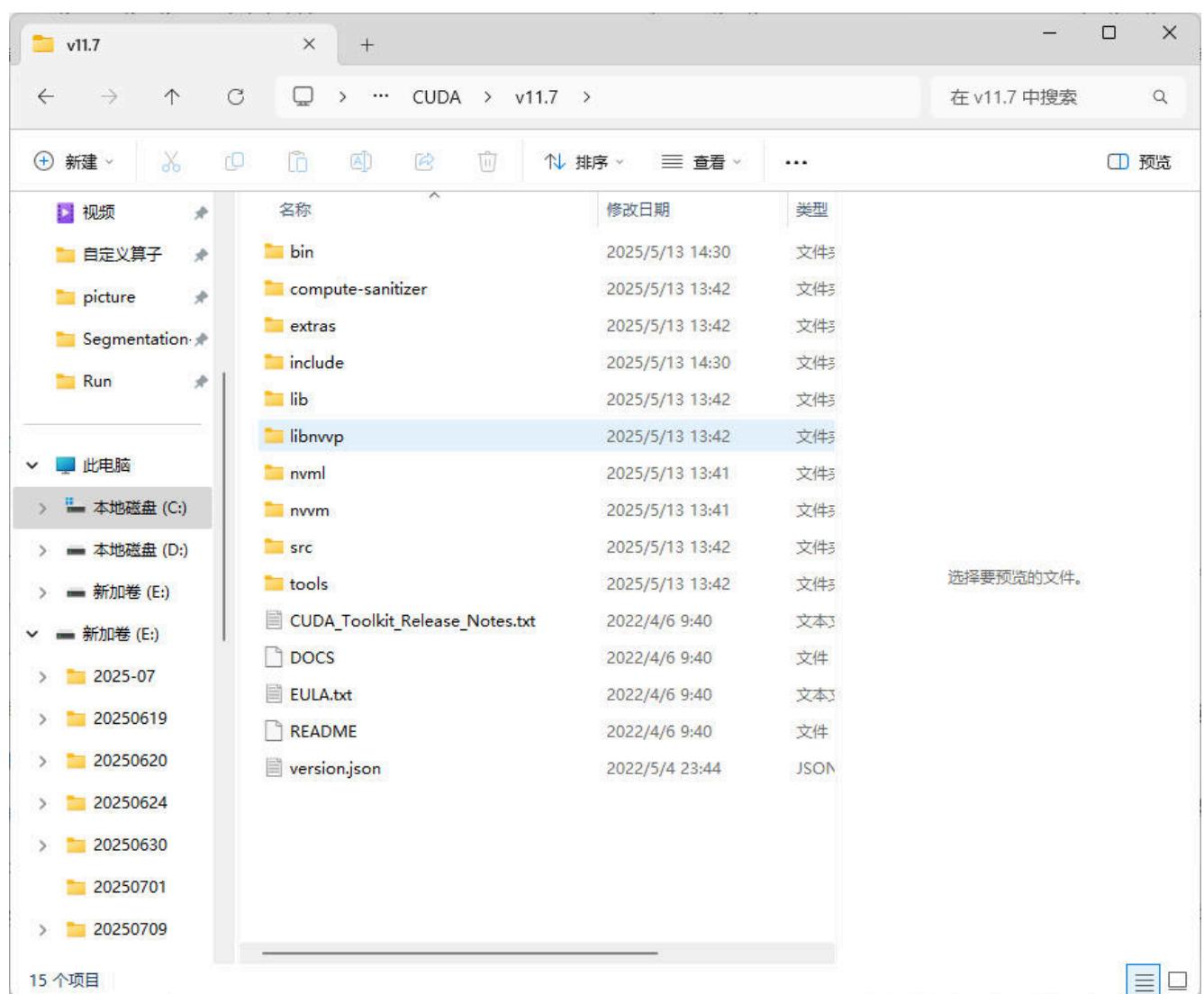
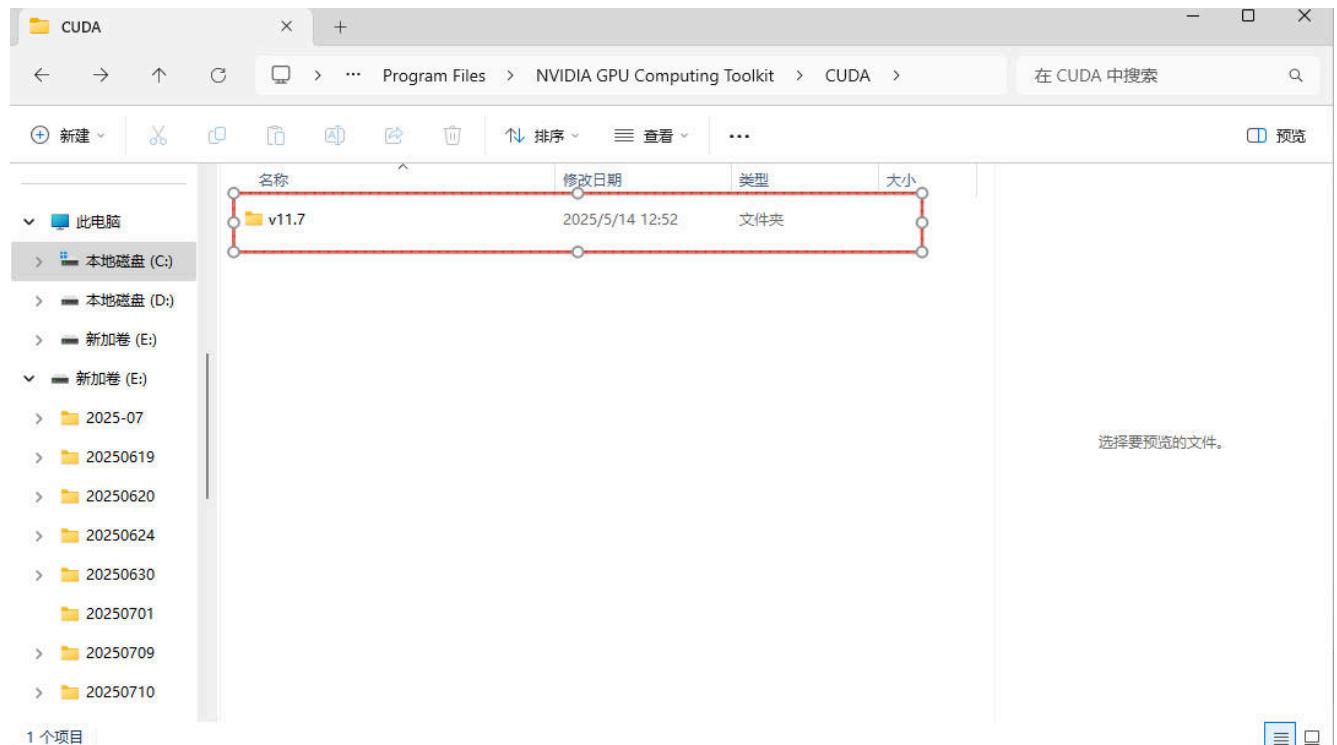


进入解压完的文件夹，复制"bin"、"include"、"lib"这三个文件夹

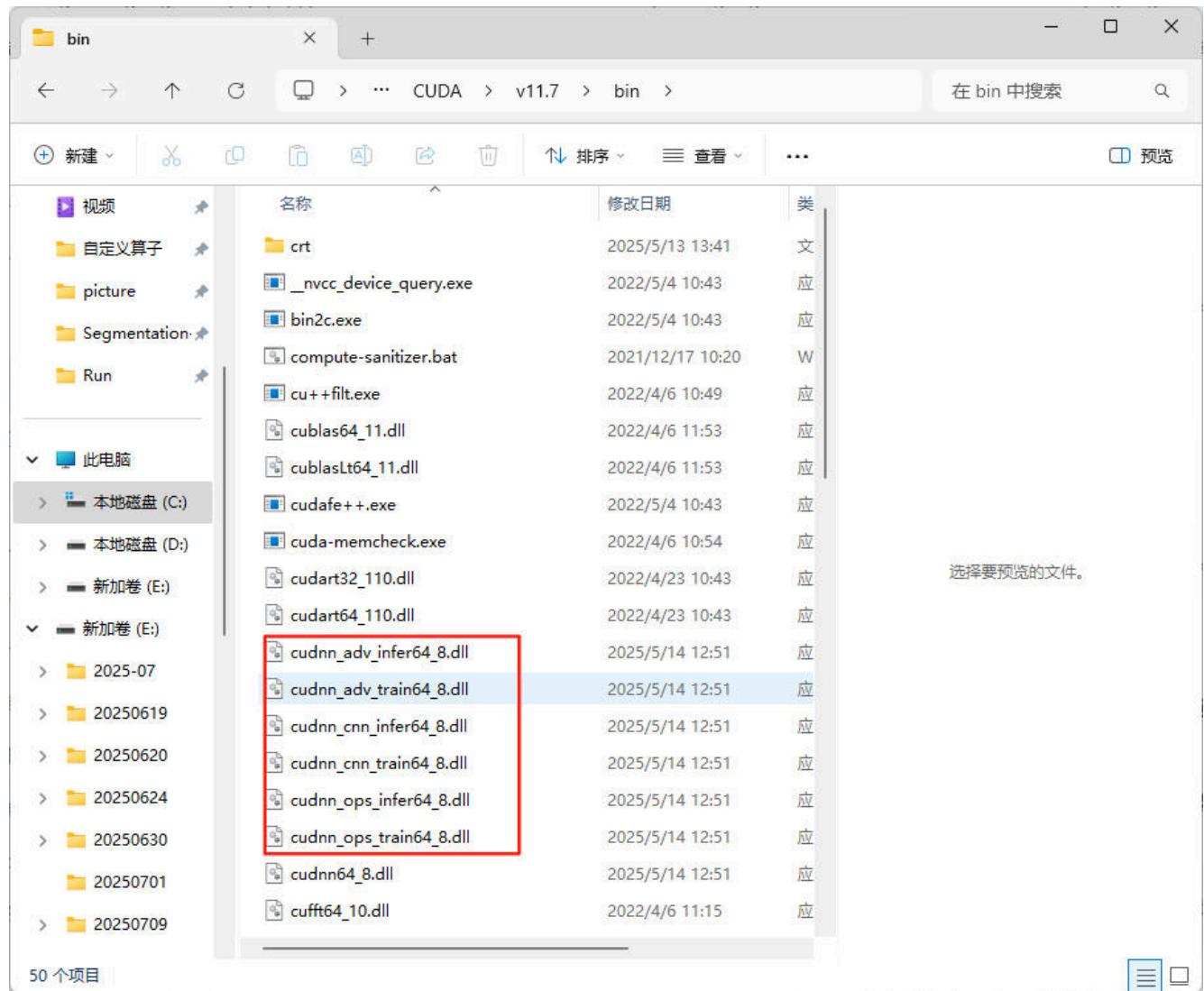


打开“C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA”里面有你安装的对应的cuda工

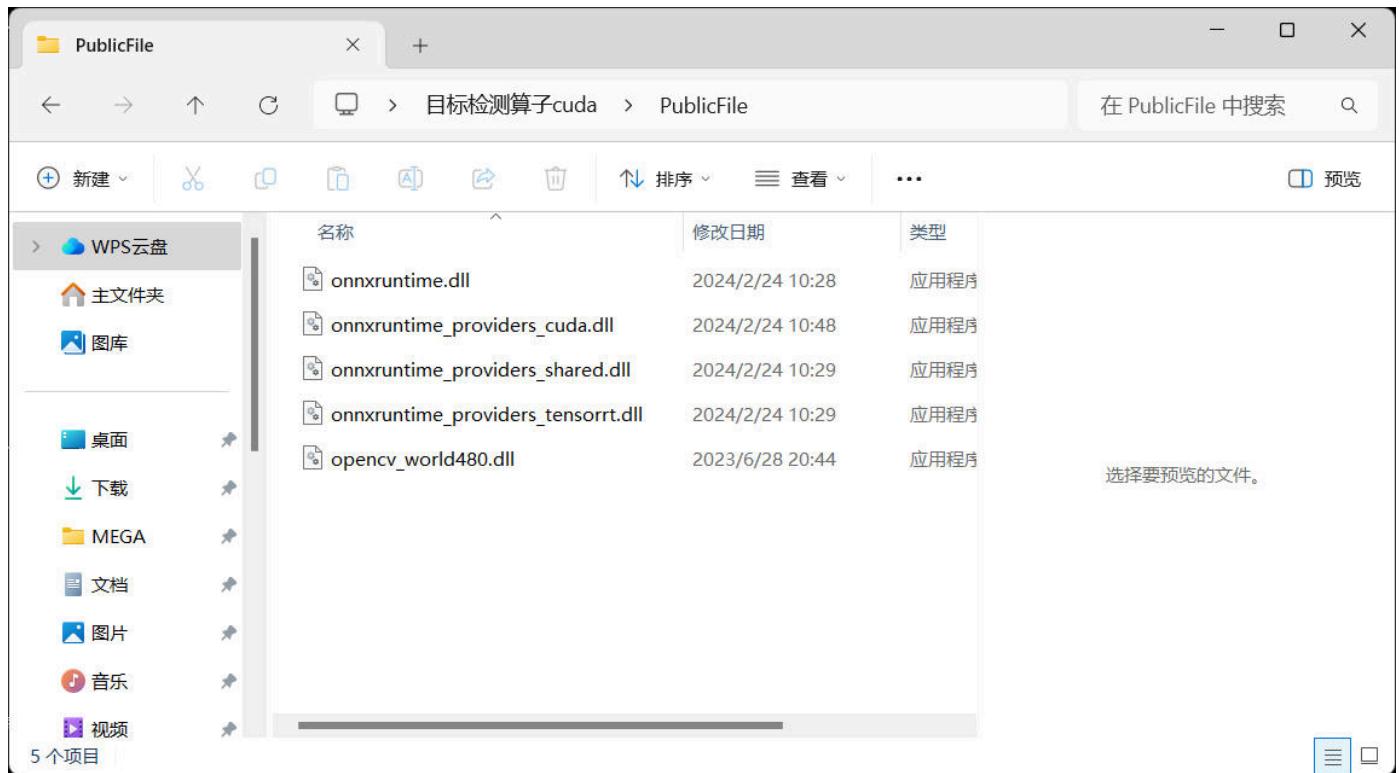
具包版本，再打开这个文件夹。



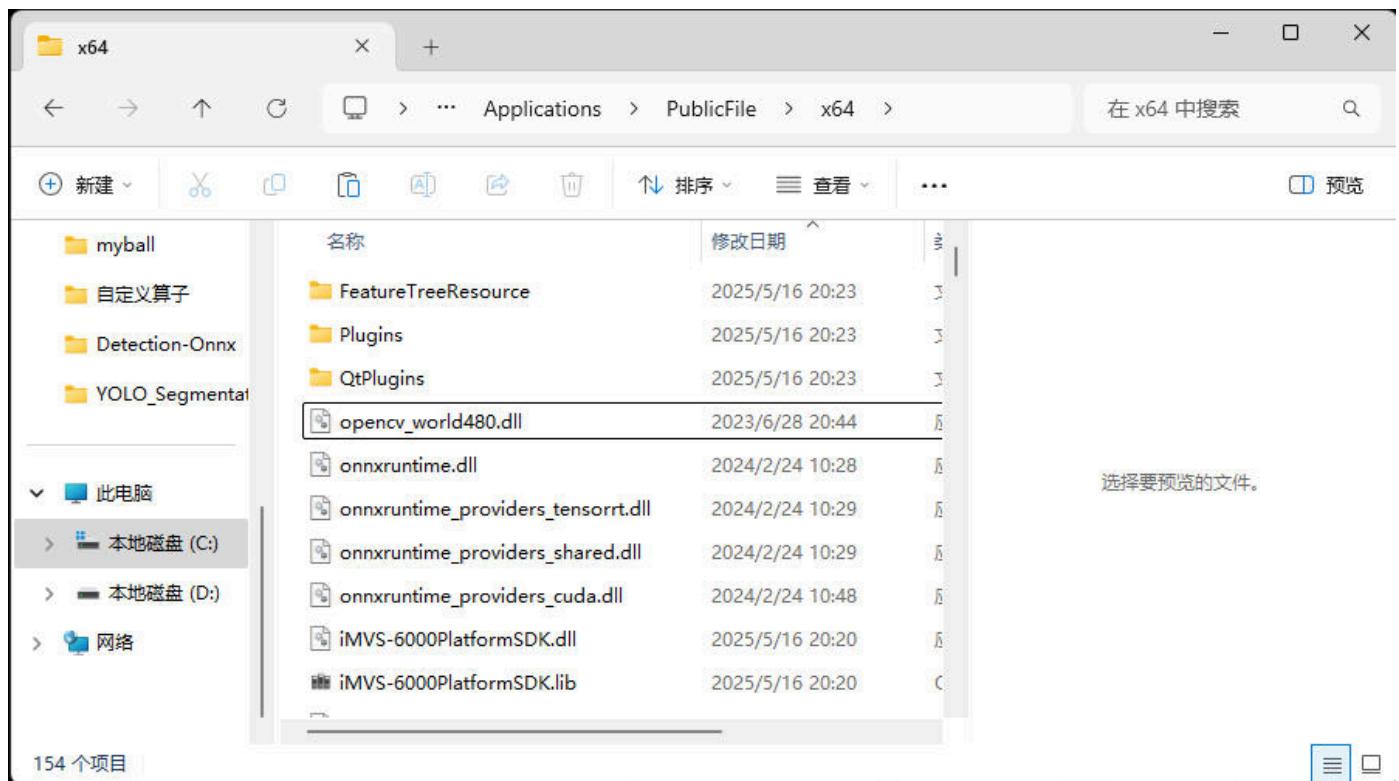
将cuDNN中复制的三个文件夹粘贴进此文件夹。粘贴完后进入"bin"中应该可以看到里面包含 cudnn



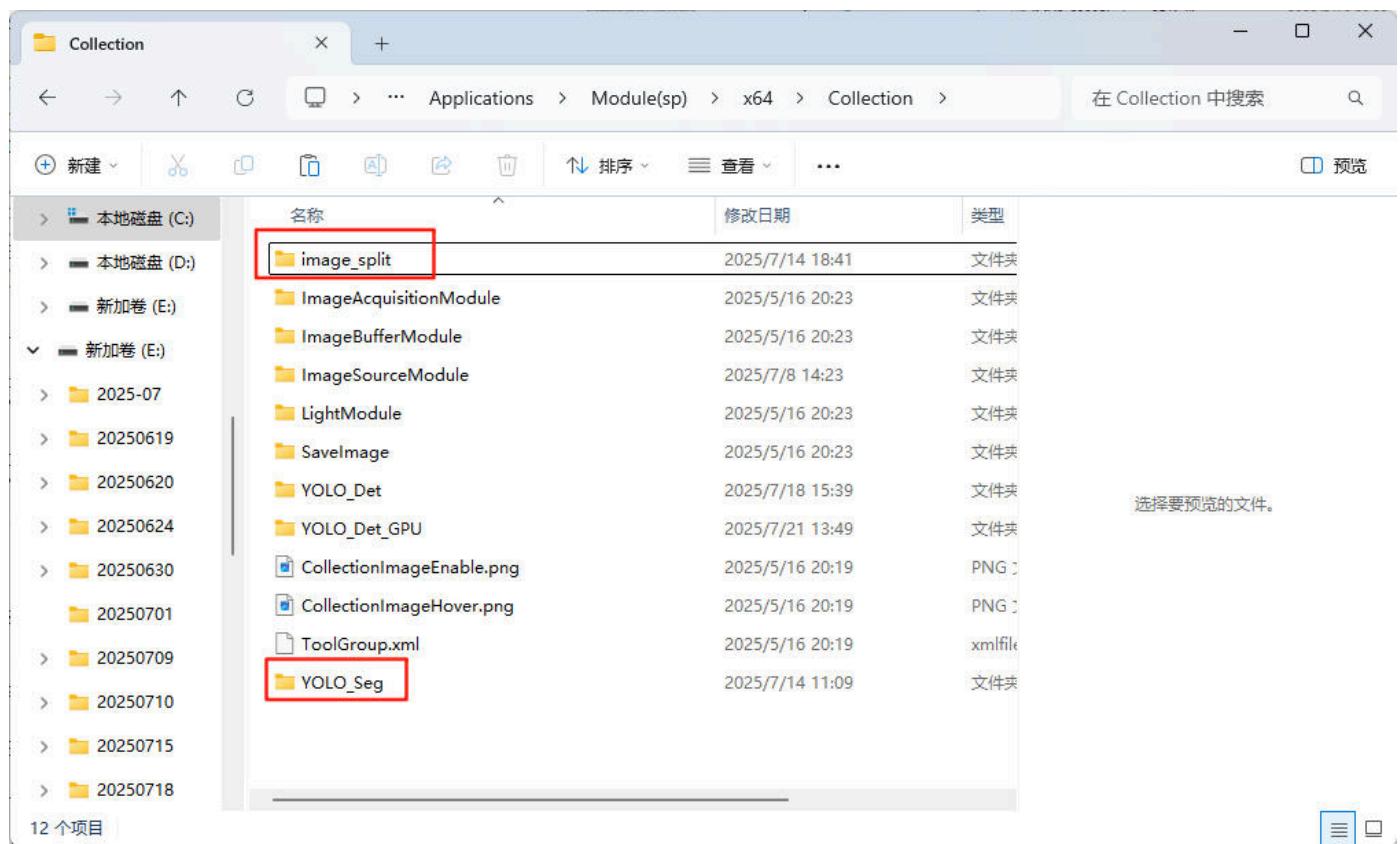
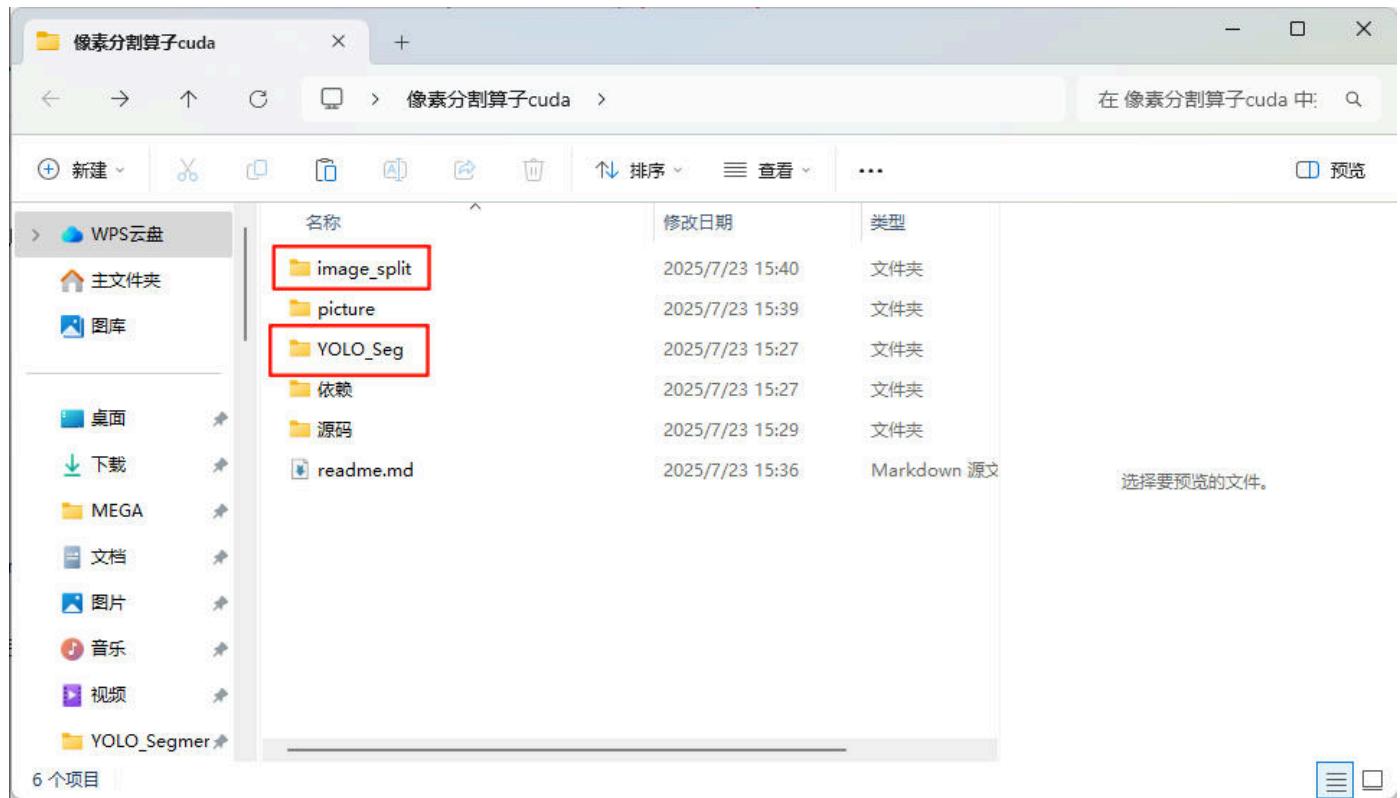
3. [下载链接](#), 下载此"PublicFile"文件, 解压后打开此"PublicFile"文件夹如下图所示



复制里面的内容，粘贴进“C盘”-“Program Files”-“VisionMaster4.4.0”-“Applications”-“PublicFile”-“x64”中



4. 复制此文档所在文件夹中的"YOLO_Seg"和"image_split",粘贴进“C盘”-“Program Files”-“VisionMaster4.4.0”-“Applications”-“Module(sp)”-“x64”-“Collection”中。



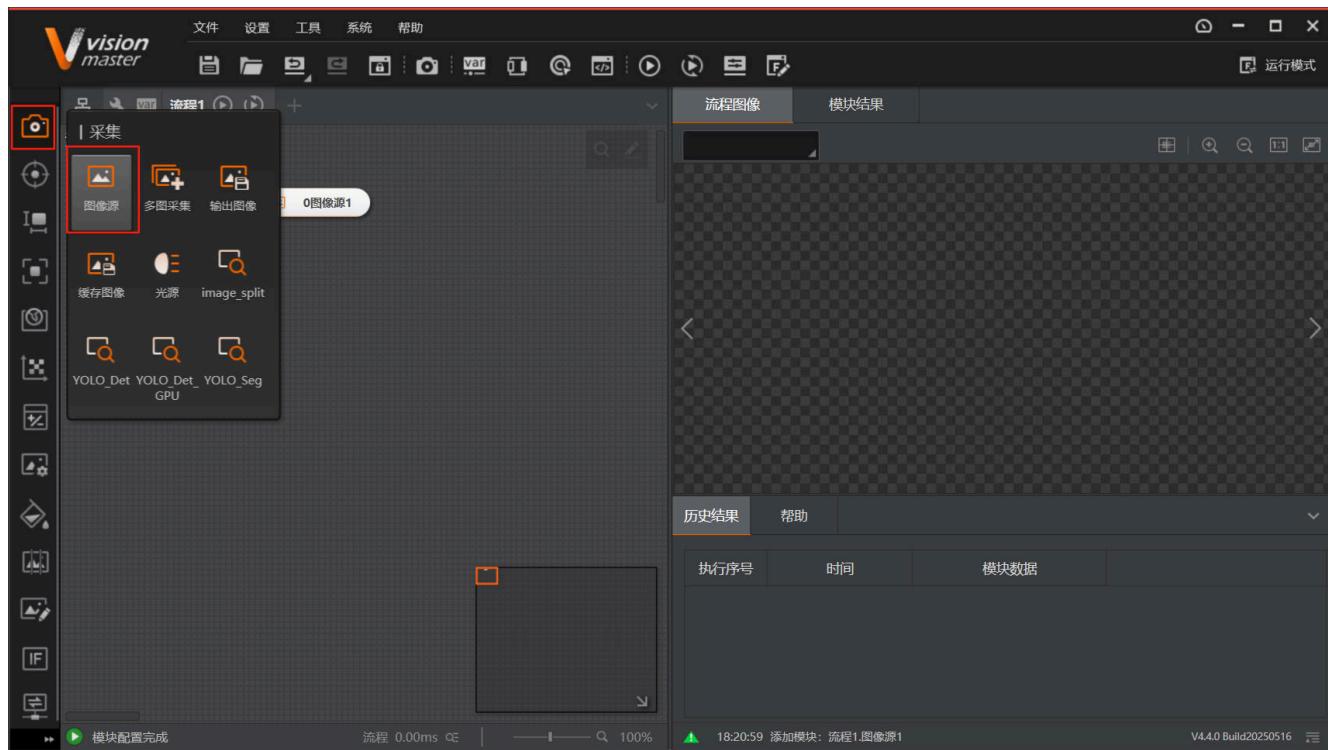
3. 快速入门

下面给出一个示例，帮助您快速了解如何使用此项目完成C++软件开发。本示例使用VS2022版本给出示例。

1. 打开"visionMaster"软件 - "通用方案"



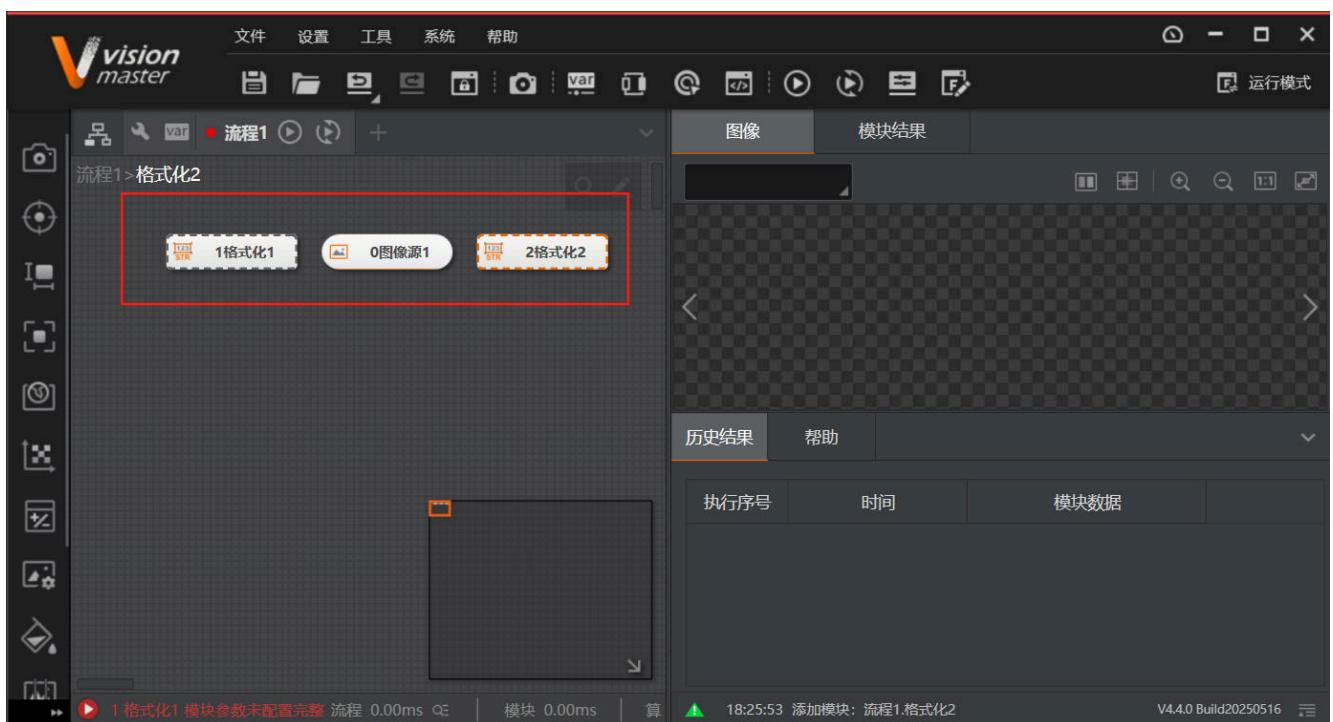
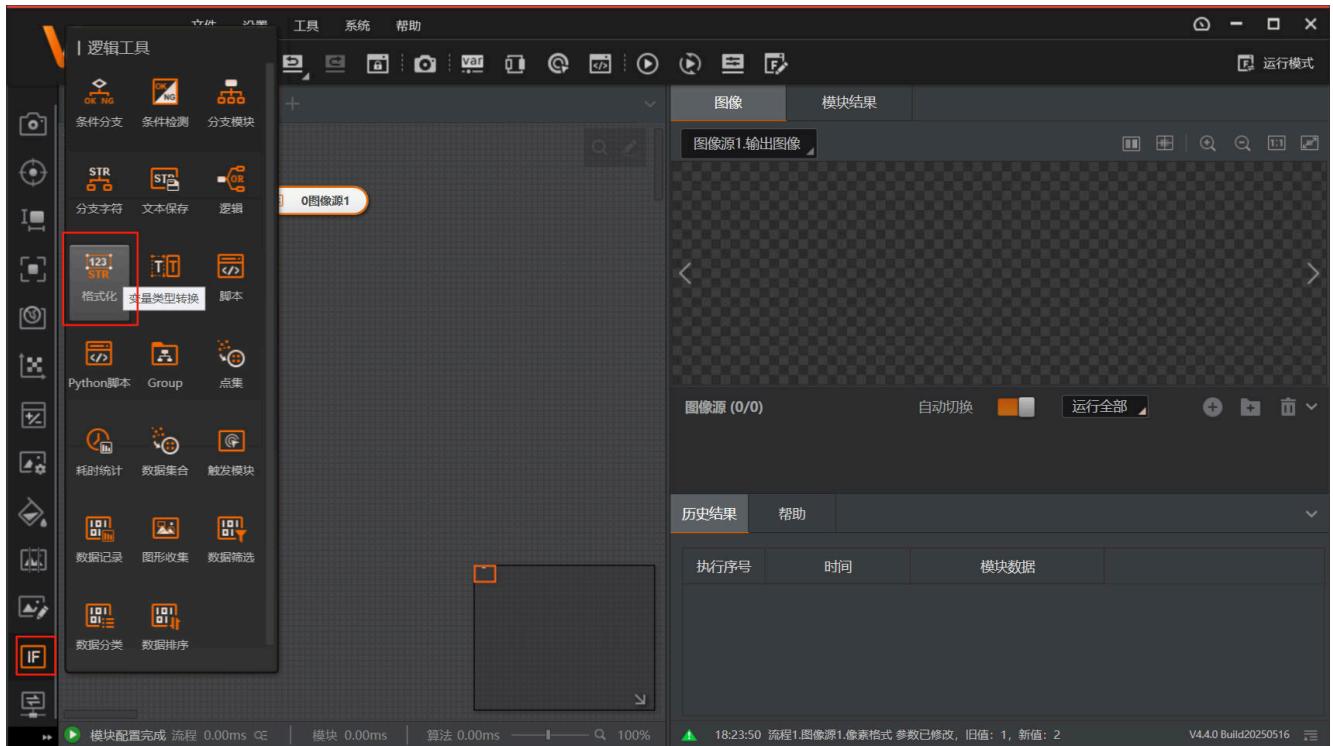
2. i. 点击"采集"- "图像源", 将其拖出至中间



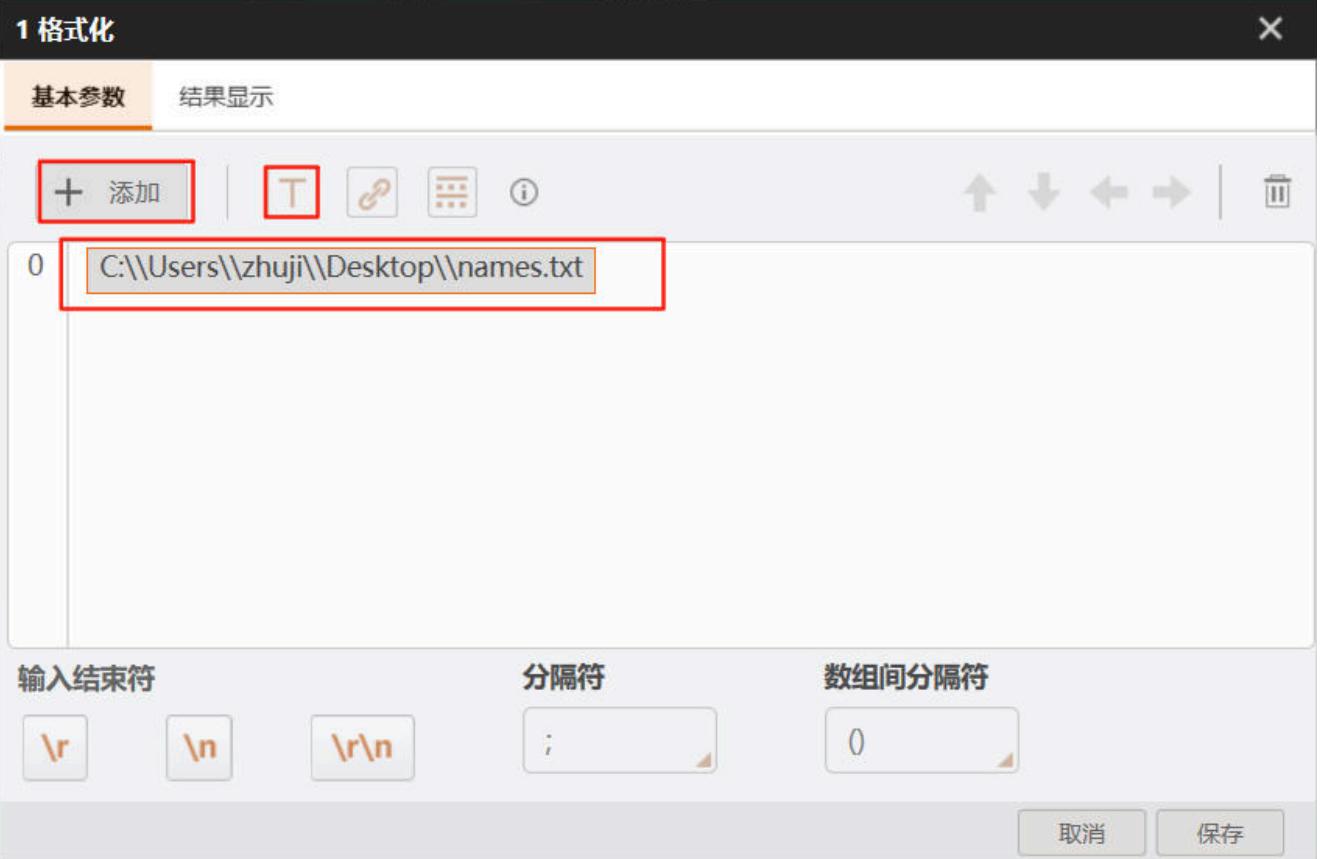
ii. 双击"0图像源1", 将像素格式修改为"RGB24"



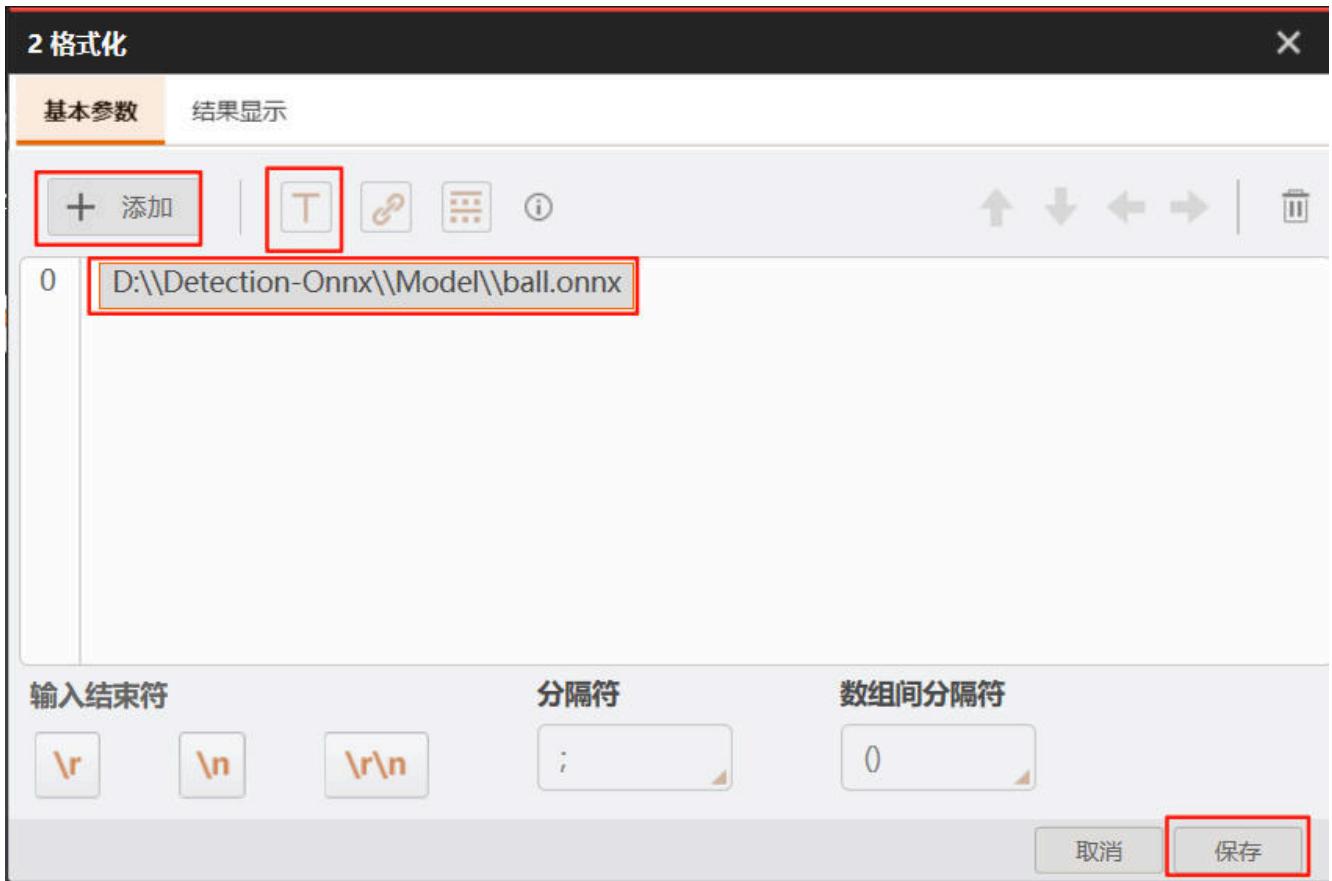
3. i. 点击“逻辑工具”-“格式化”，拖出两个



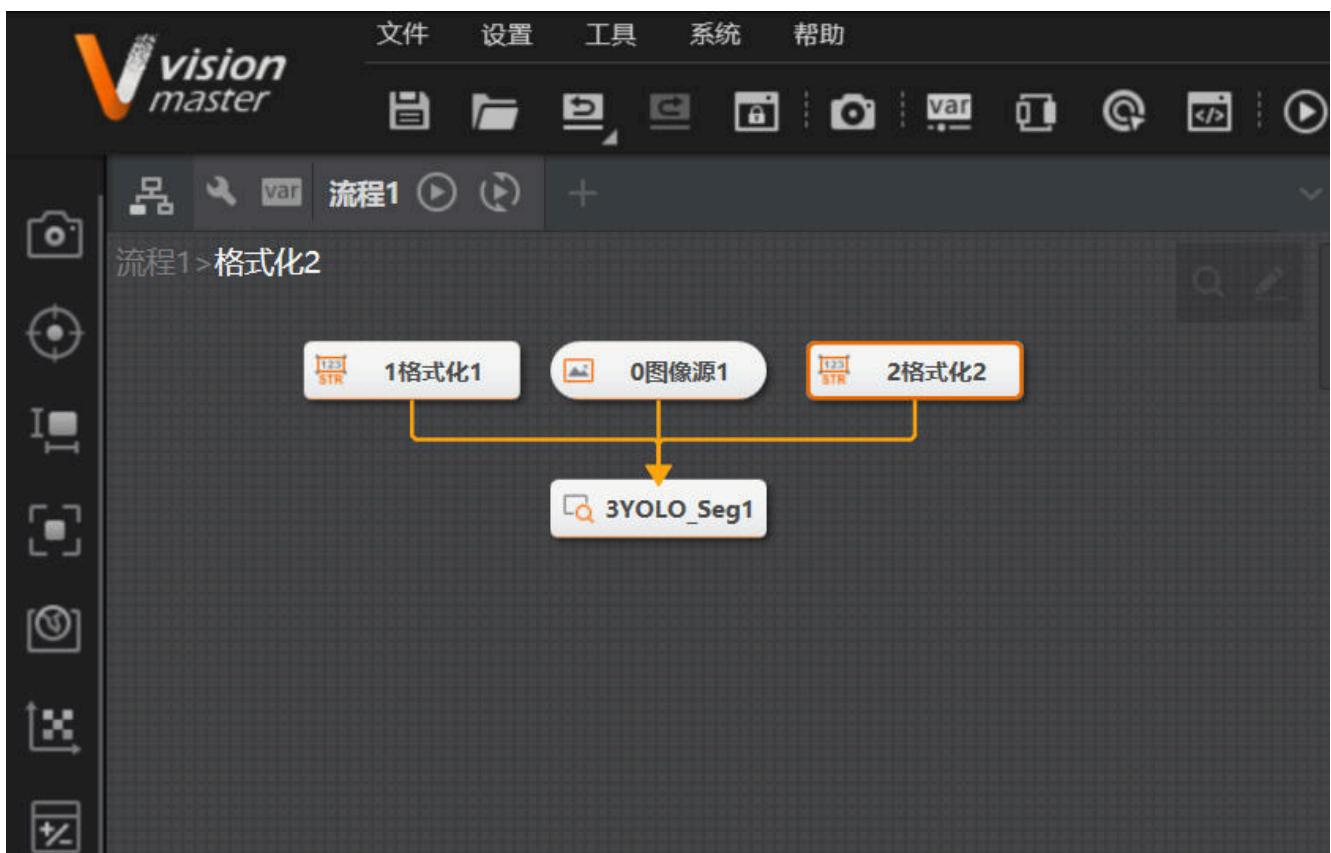
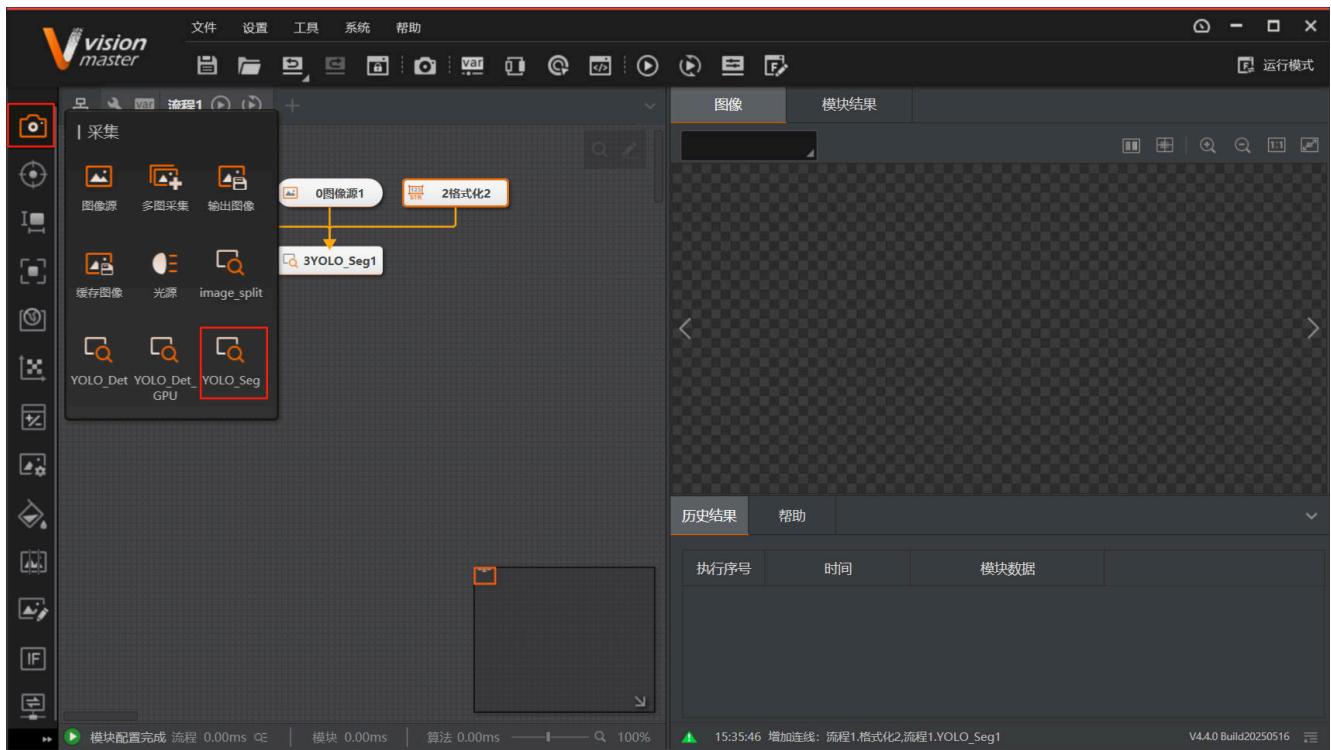
- ii. 双击"1格式化1"，点击"添加"，"T"，在下面的行中输入“标签路径”(names.txt所在路径，names.txt需要用utf-8保存)，然后点击右下角“保存”



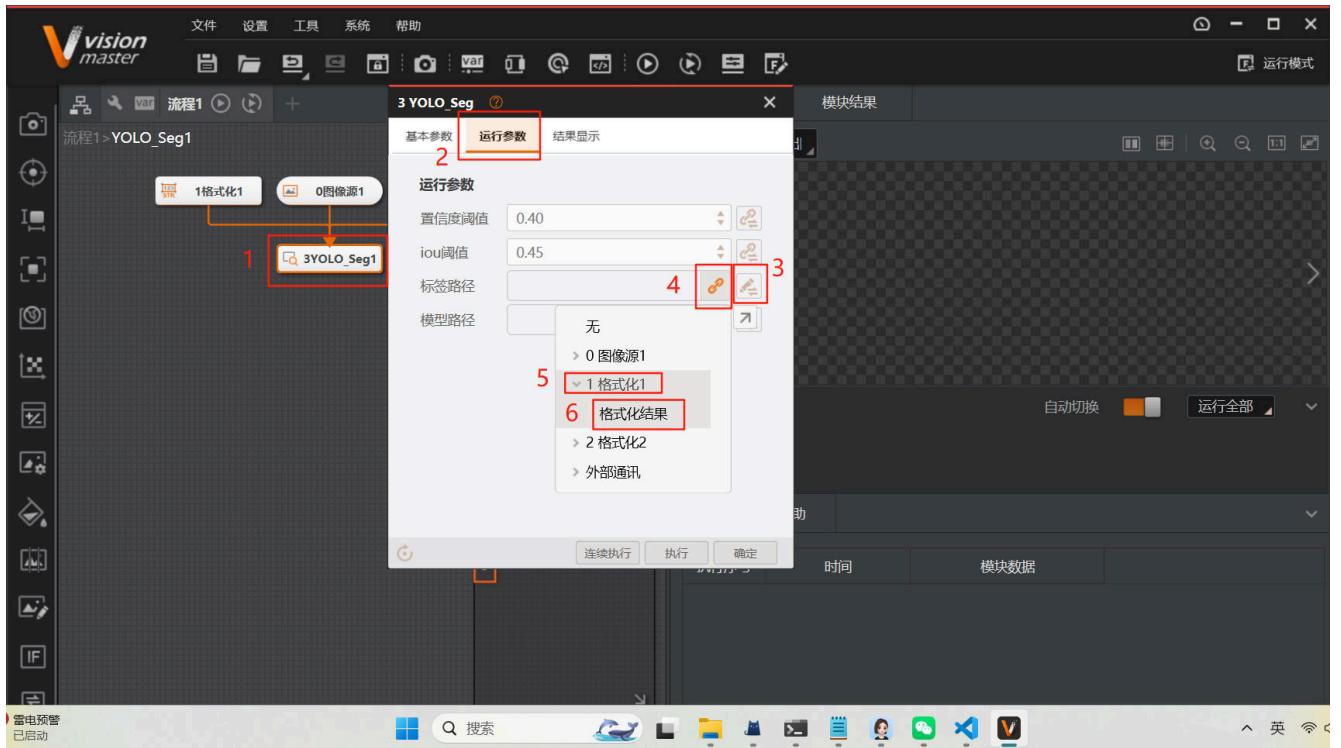
- iii. 双击"2格式化2", 点击"添加", "T", 在下面的行中输入“模型路径”, 然后点击右下角保存



4. i. 点击“采集”-“YOLO_Seg”,将其拖出, 按照如图所示连接



- ii. 双击"3YOLO_Seg1",点击"运行参数"->标签路径右侧的按钮"->该按钮左侧的链接按钮"->"1格式化1"->"格式化结果"



iii. 点击“模型路径右侧的按钮”-“该按钮左侧的链接按钮”-“2格式化2”-“格式化结果”

基本参数

运行参数

结果显示

运行参数

置信度阈值

0.40



iou阈值

0.45



标签路径

1 格式化1.格式化结果[]



模型路径



> 0 图像源1



▼ 1 格式化1

格式化结果

2 格式化2

格式化结果

> 外部通讯



连续执行

执行

确定

iv. 点击右下角的"确定"

基本参数

运行参数

结果显示

运行参数

置信度阈值

0.40



iou阈值

0.45



标签路径

1 格式化1.格式化结果[]



模型路径

2 格式化2.格式化结果[]

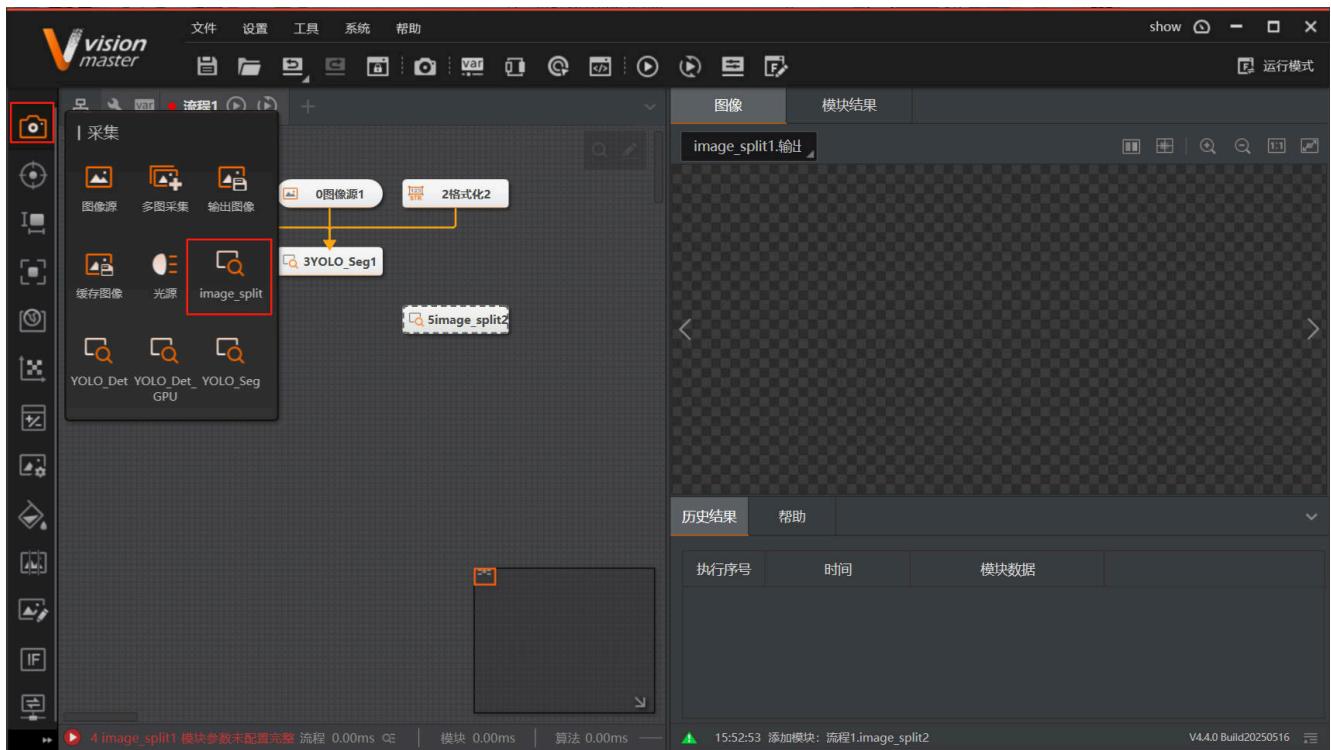


连续执行

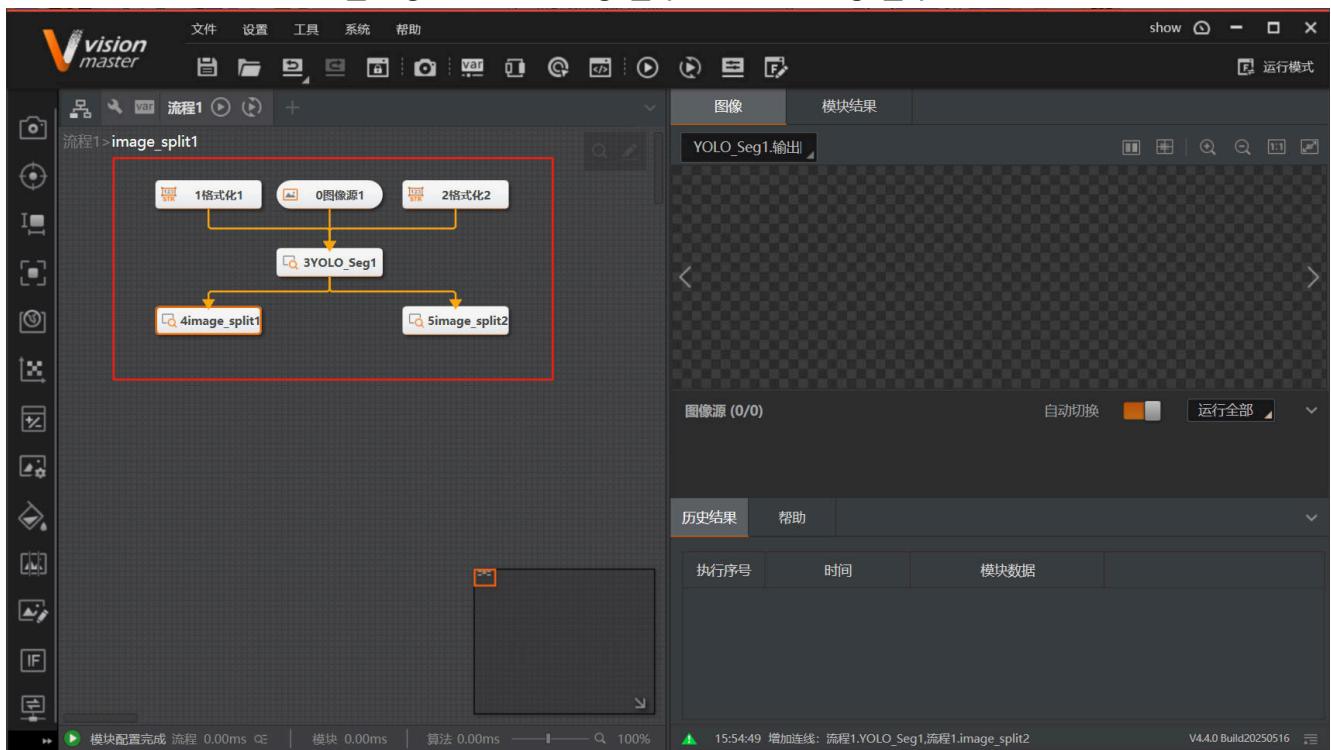
执行

确定

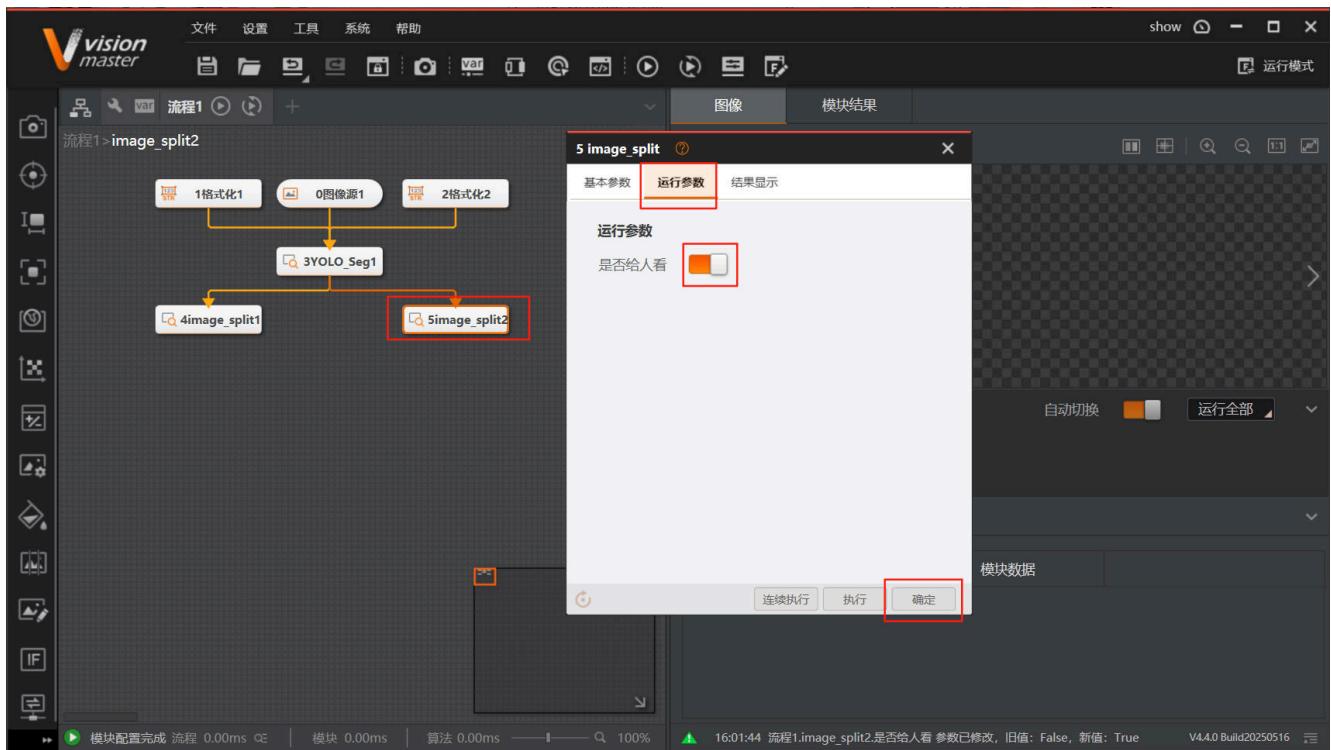
5. i. 点击"采集"->"image_split",拖出两个"image_split"



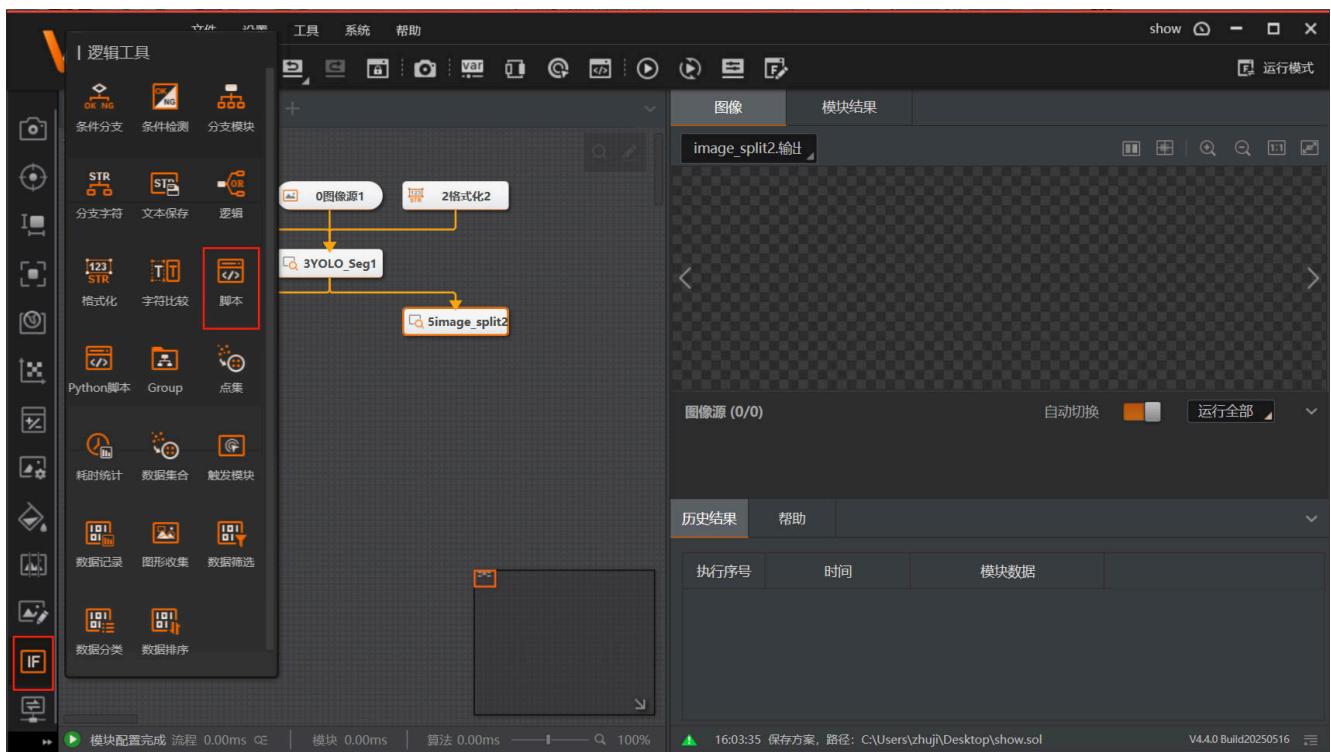
按原图所示，将“3YOLO_Seg1”连接“4image_split1”和“5image_split2”如下图所示



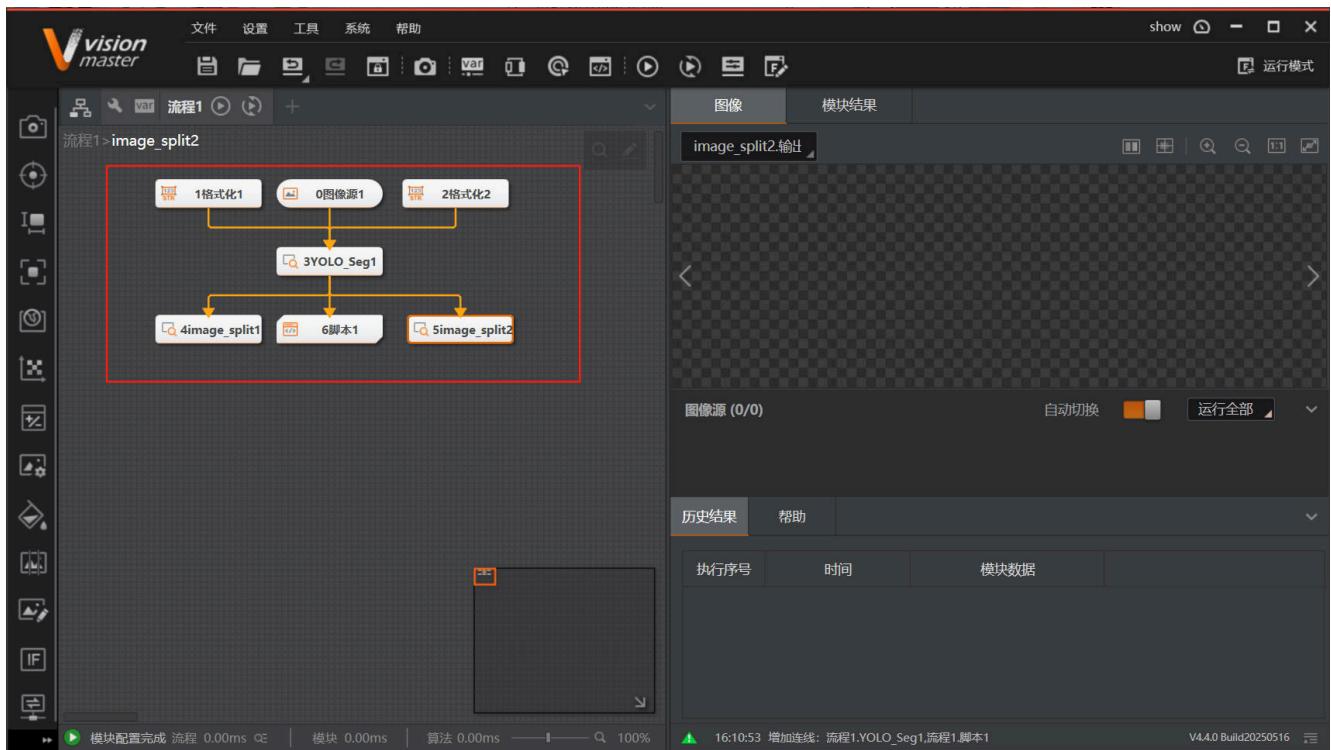
- ii. 双击"5image_split2",点击"运行参数" - "是否给人看"，使这个选项处于打开状态，再点击右下角的"确定"



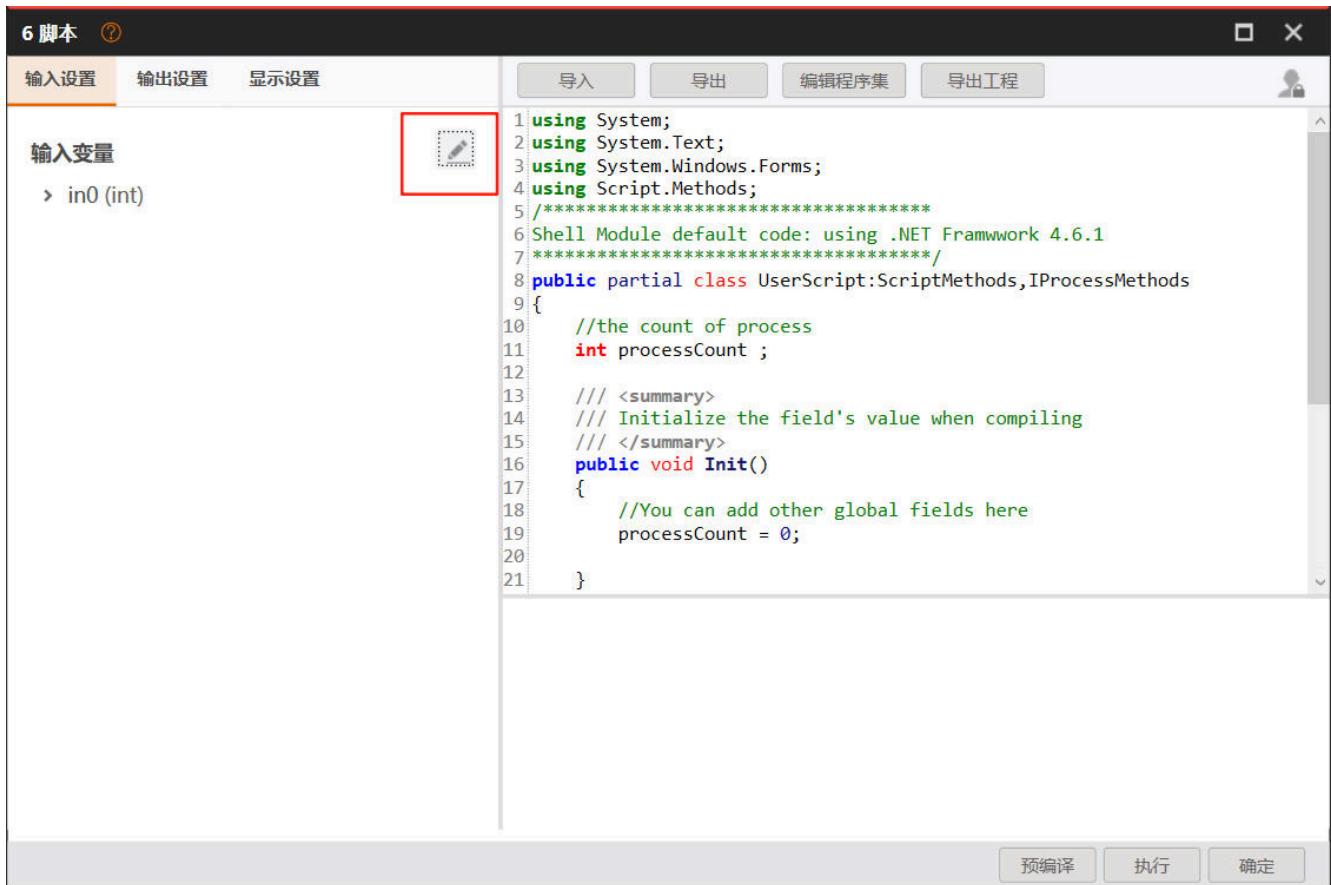
6. i. 点击"逻辑工具"-“脚本”，将其拖出。

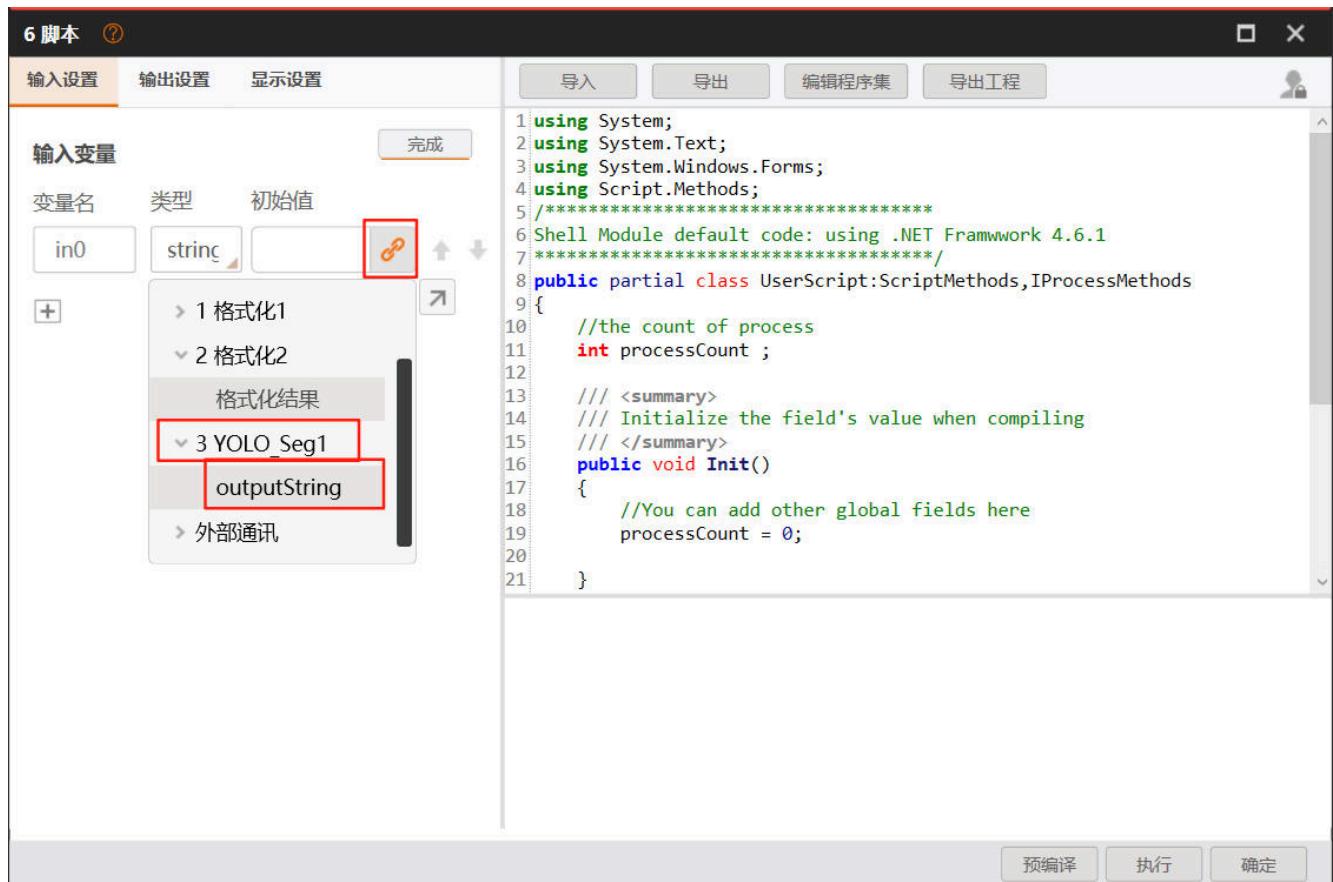
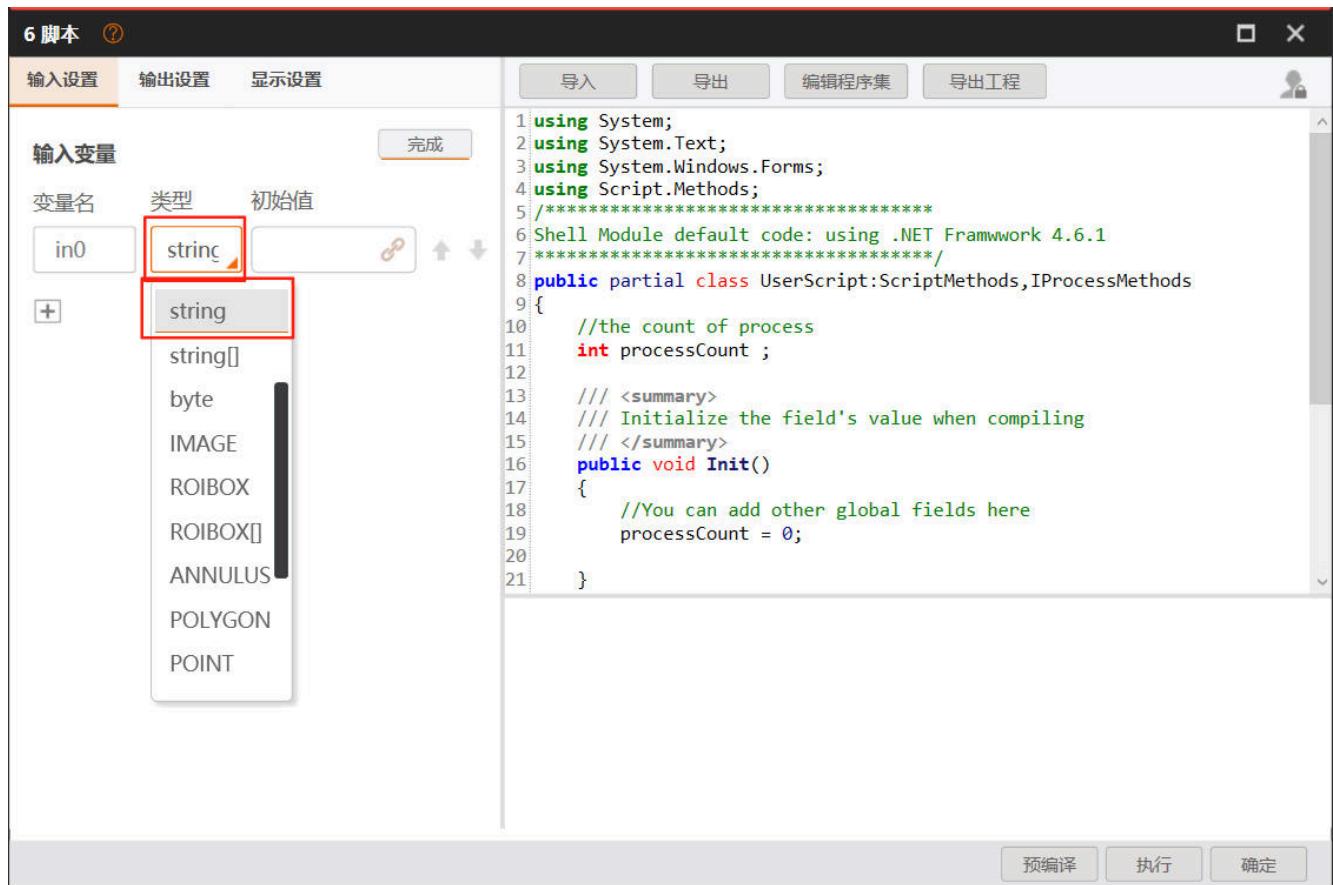


按下图所示，将"3YOLO_Seg1"与"6脚本1"连接



- ii. 双击打开"6脚本1", 点击"输入变量右侧的铅笔", 将"类型"切换成"string", 在点击"初始值处的按钮"->"3YOLO_Det1"->"outputString"





点击右上角"完成"

6 脚本

输入设置 输出设置 显示设置 导入 导出 编辑程序集 导出工程 个人

完成

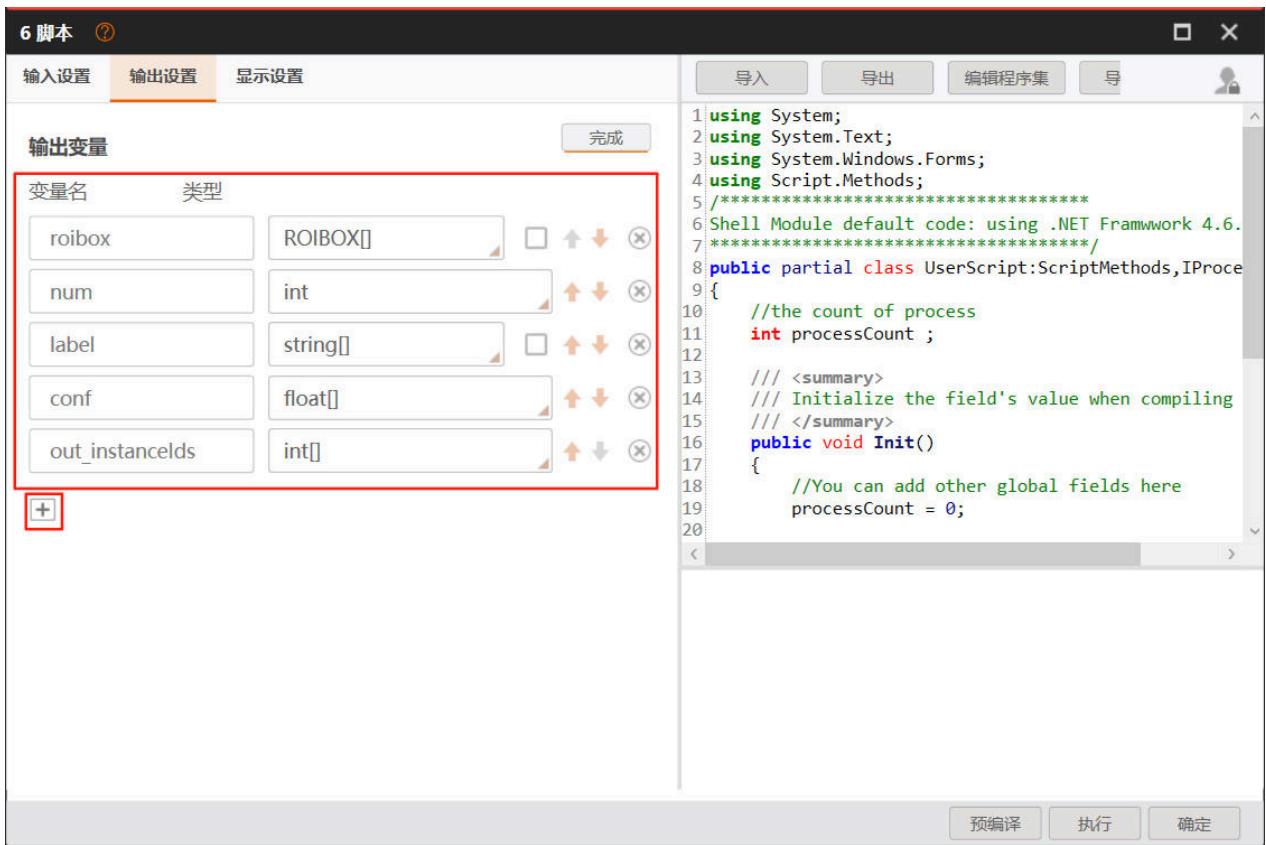
变量名	类型	初始值
in0	string	3 YOLO_Seg1.outputStr

+ 1 **using** System;
2 **using** System.Text;
3 **using** System.Windows.Forms;
4 **using** Script.Methods;
5 *********
6 Shell Module default code: **using** .NET Framework 4.6.1
7 *********
8 **public partial class** UserScript:ScriptMethods,IProcessMethods
9 {
10 **//the count of process**
11 **int** processCount ;
12 **/// <summary>**
13 **/// Initialize the field's value when compiling**
14 **/// </summary>**
15 **public void** Init()
16 {
17 **//You can add other global fields here**
18 processCount = 0;
19 }
20 }
21 }

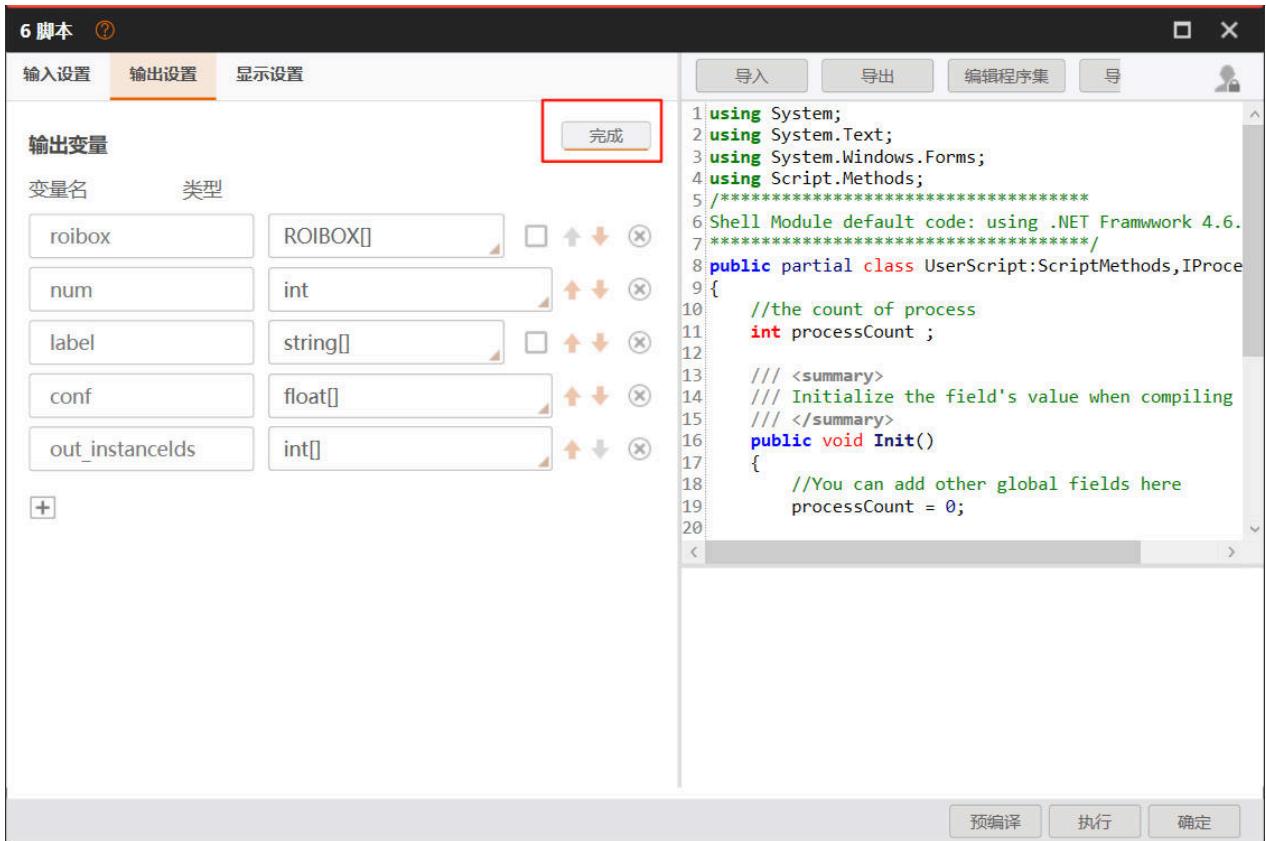
预编译 执行 确定

iii. 点击"输出设置",点击"输出变量右侧的铅笔",修改成如图所示,增加一行是按红框中的"+"
每行的内容分别是

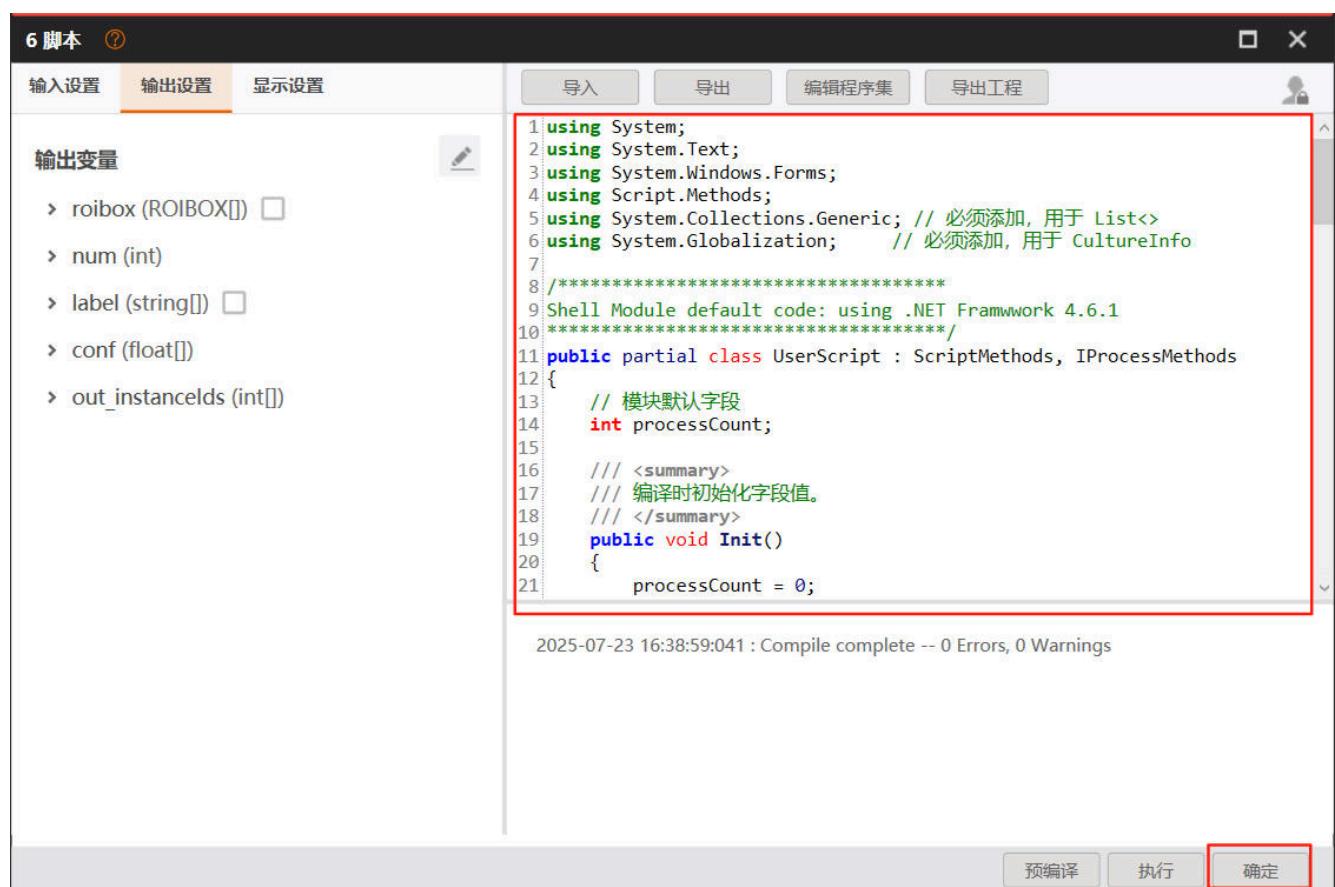
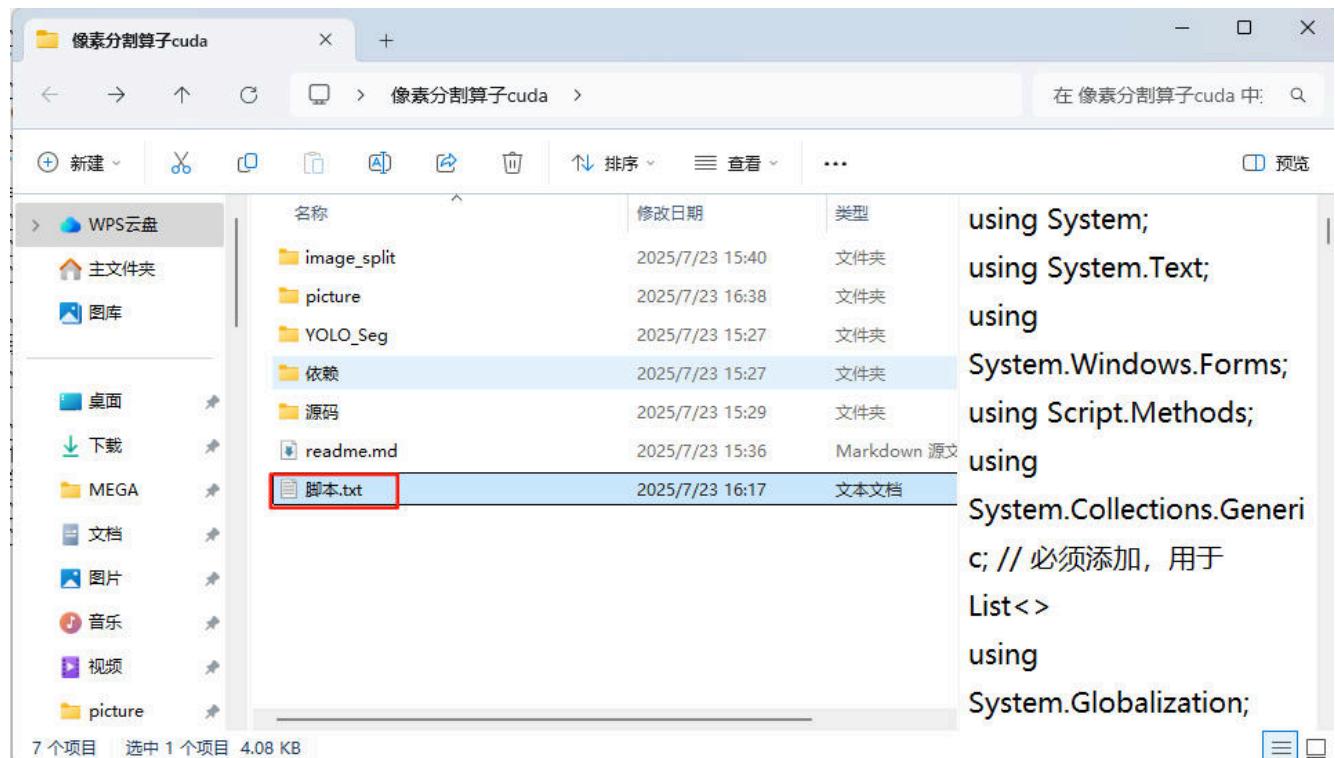
- a. roibox-ROIBOX[]
- b. num-int
- c. label-string[]
- d. conf-float[]
- e. out_instanceIds-int[]



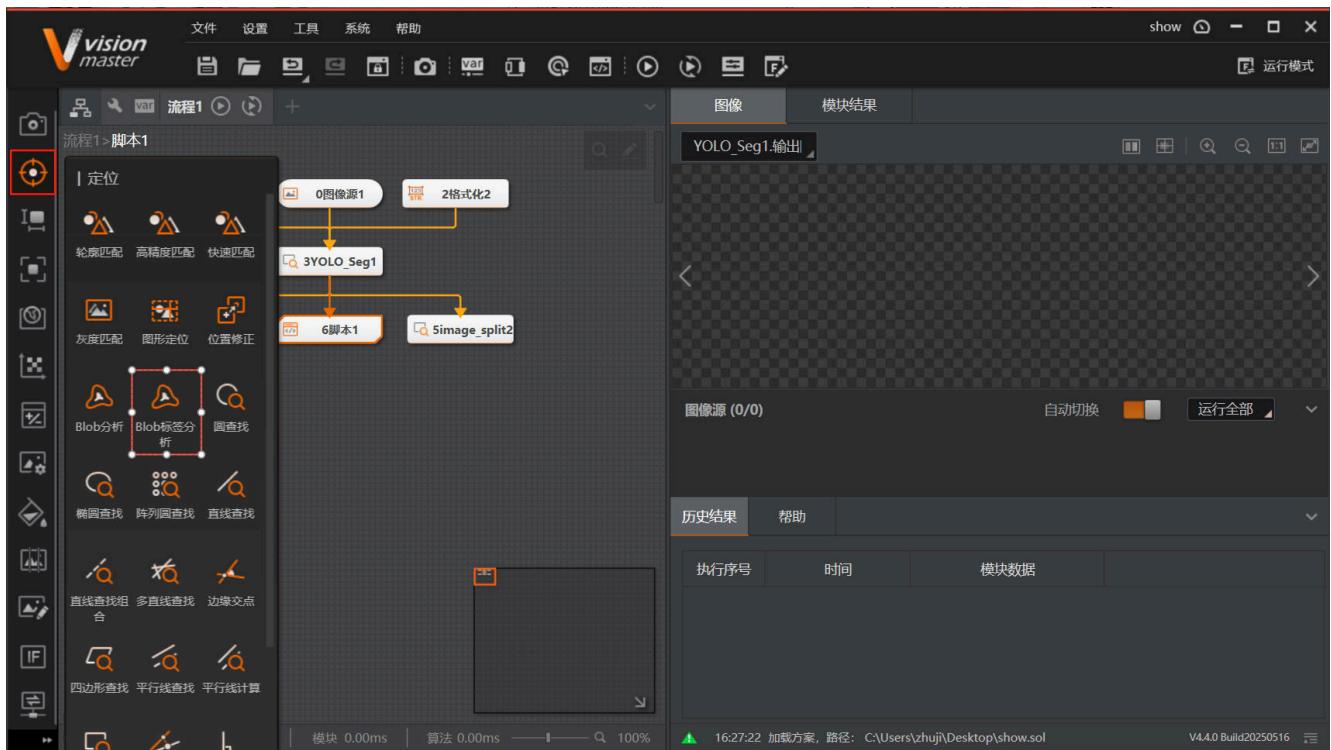
点击右上角"完成"



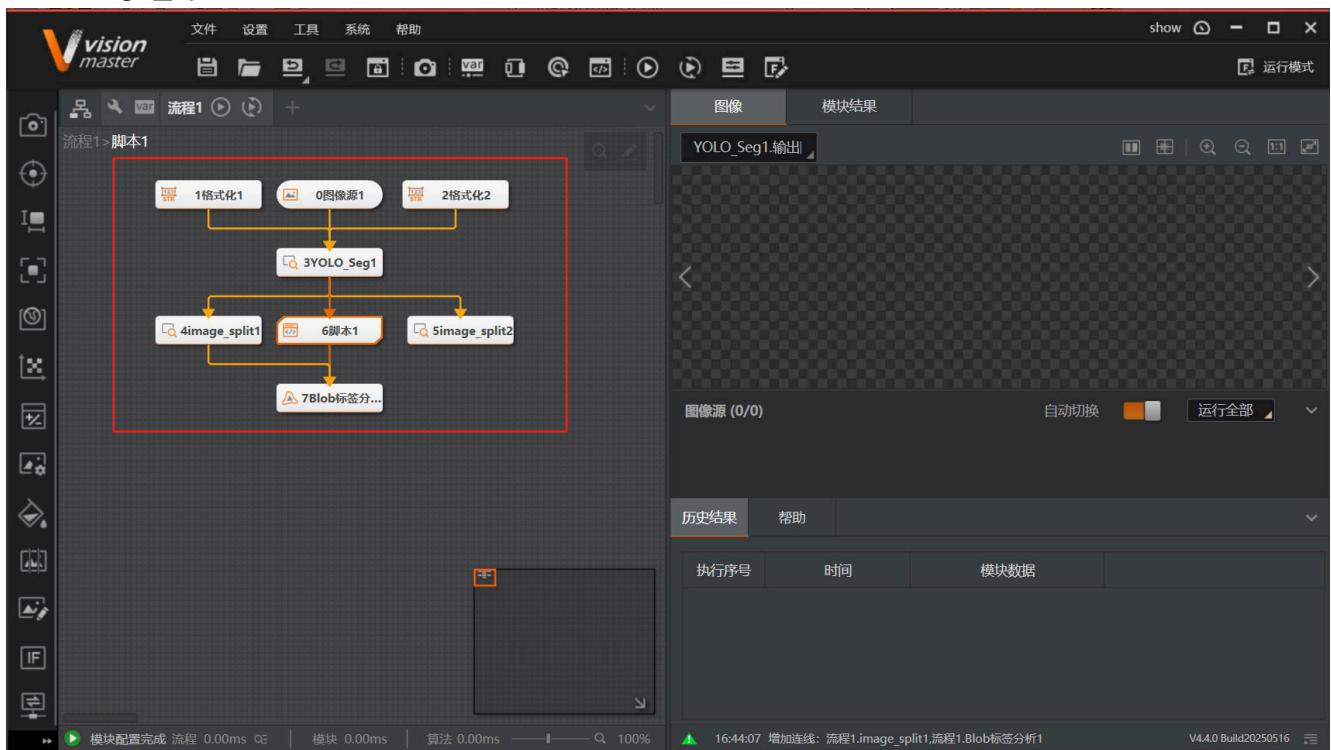
- iv. 打开此文档所在文件夹中的"脚本.txt",复制其中的内容, 粘贴进visionMaster4.4中的"6脚本"的右侧, 替换其中的原有代码



7. i. 点击"定位"->"Blob标签分析"



将"4image_split1"、"6脚本1"连接到"7Blob标签分析1"上



ii. 将图像输入-输入源修改成"4image_split1 1.输出图像"

7 Blob标签分析



基本参数

运行参数

结果显示

图像输入

输入源

4 image_split1.输出图像

ROI区域

输出掩膜

4 image_split1.输出图像

ROI区域类型

3 YOLO_Seg1.输出图像

ROI创建

0 图像源1.图像

0 图像源1.灰度图像

ROI创建

绘制 继承

形状



屏蔽区创建

绘制 继承

屏蔽区



位置修正

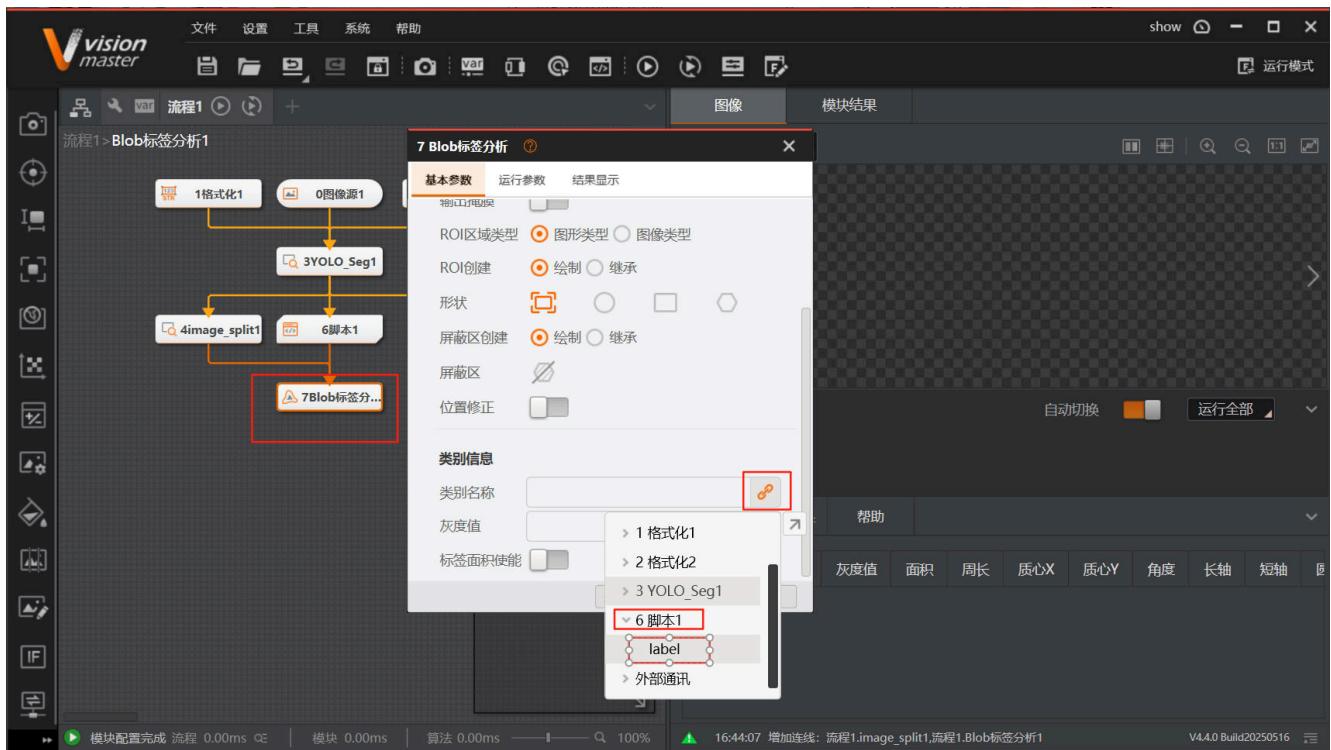


连续执行

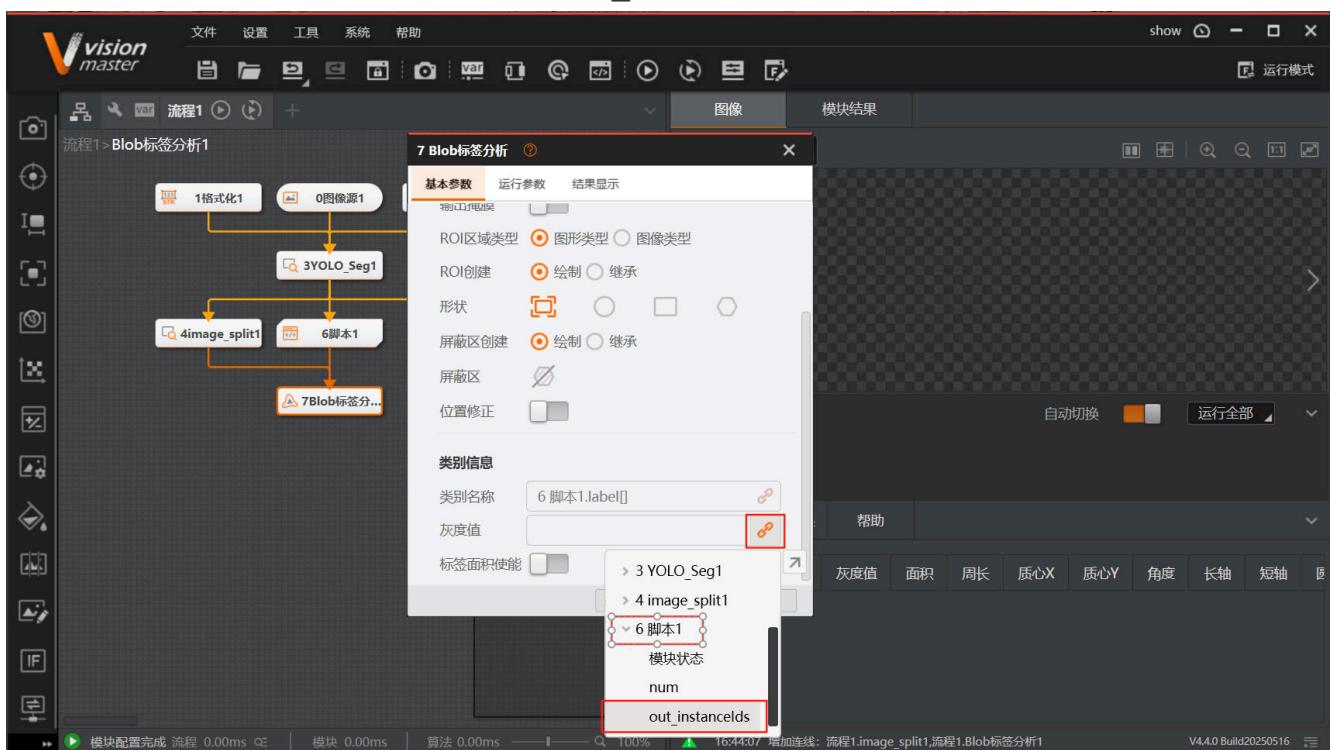
执行

确定

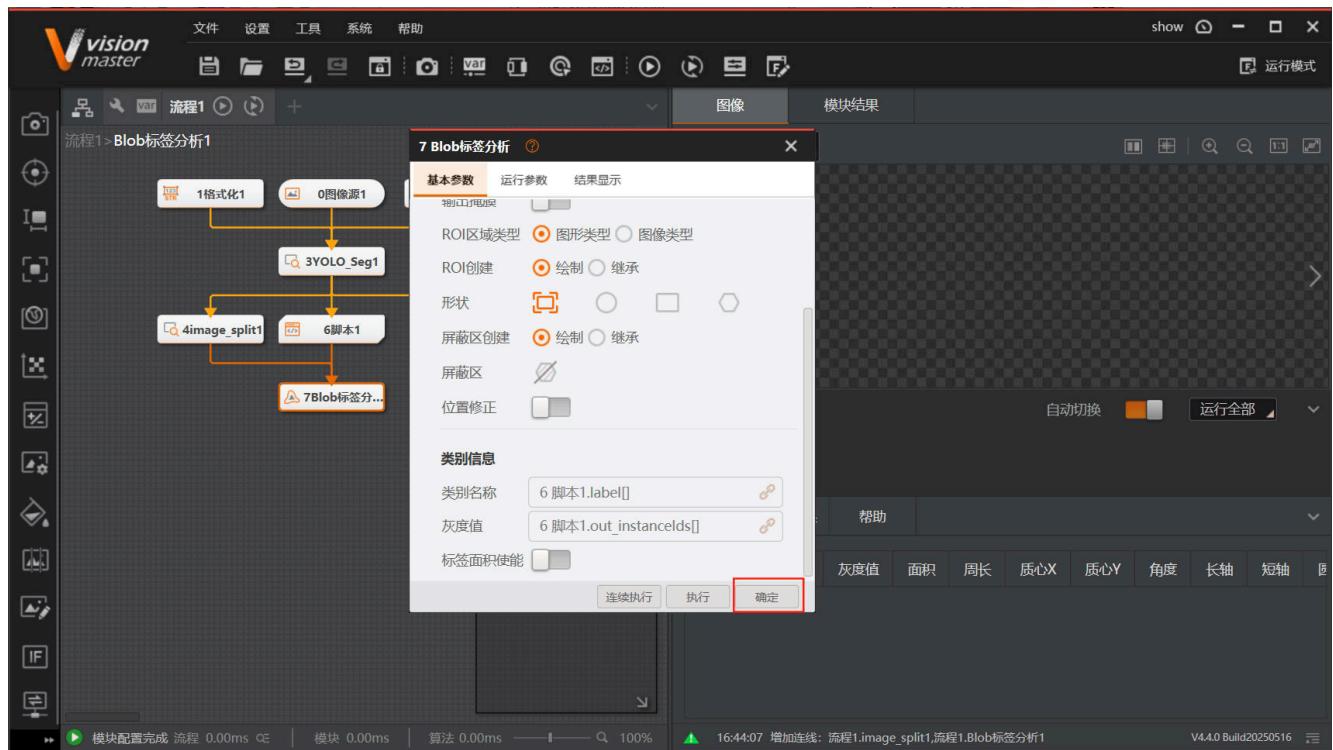
iii. 双击"7Blob标签分析1",点击"类别名称右侧的链接按钮"->"6脚本1"->"label"



iv. 点击"灰度值右侧的链接按钮"->"6脚本1"->"out_instanceIds"



v. 点击右下角的"确定"



8. 此时运行"5image_split2"中显示人眼可见分割结果, "7Blob标签分析"中可以看见blob标签分析的结果

