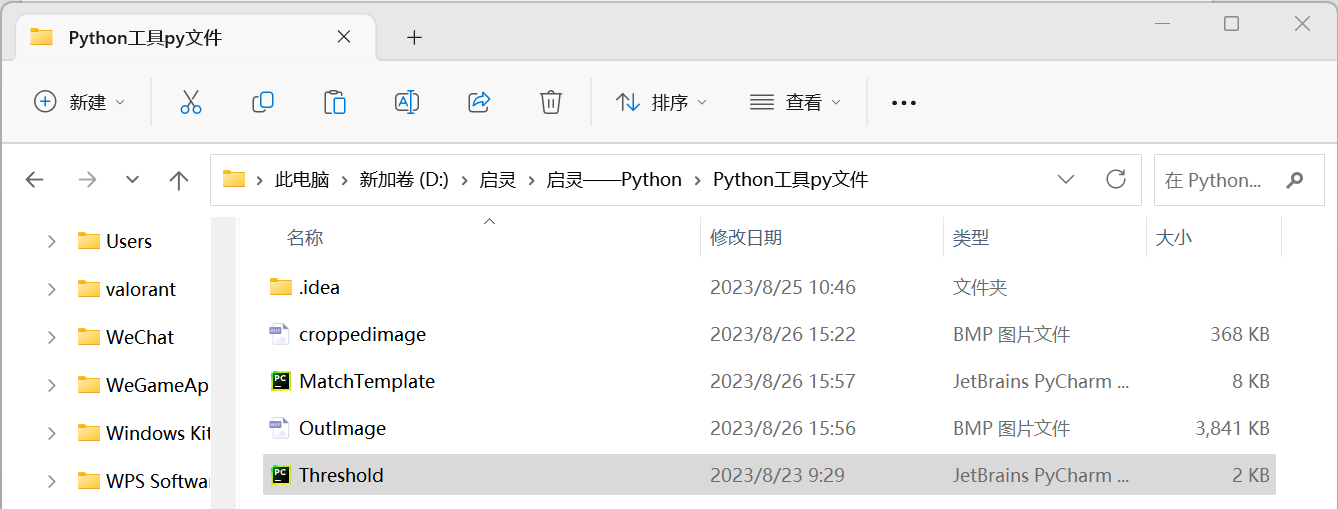
# 1.1阈值化

用于从一幅图像中，利用阈值分割出我们需要物体的部分。

1.1.1基础参数设置

脚本路径设置：双击脚本路径框，找到并打开相应的.py文件。如图1.1.1所示。

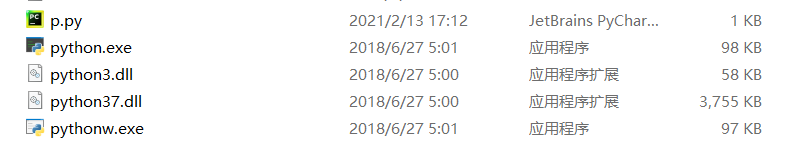




**图1.1.1 脚本参数设置**

解释器路径设置：双击解释器路径框，找到并打开相应的python.exe应用程序。如图1.1.2所示。

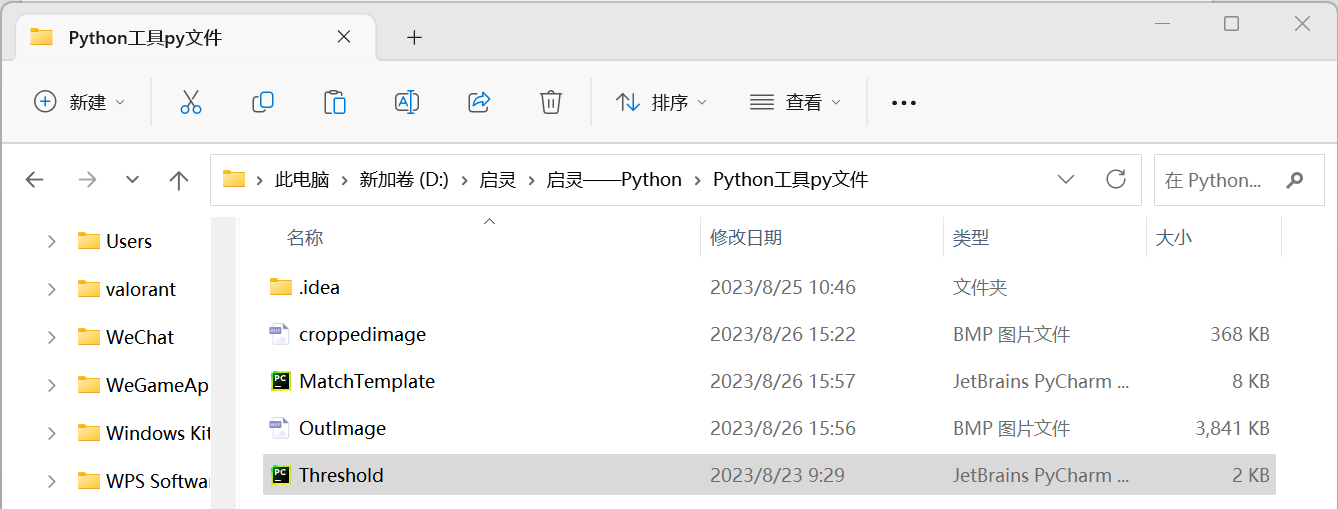
****



**图1.1.2 解释器路径设置**

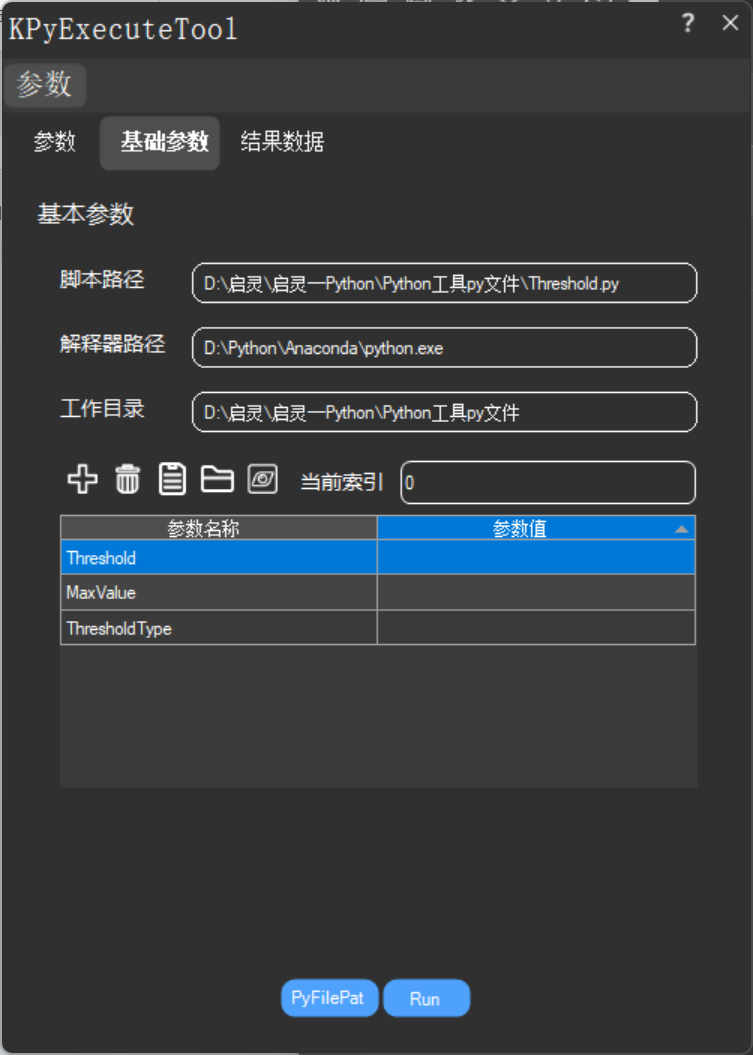
工作目录设置：双击工作目录框，找到并打开相应的.py文件（或直接复制脚本路径最后一个斜杠前的路径粘贴到此处）。如图1.1.3所示。





**图1.1.3 工作目录设置**

添加参数：点击三次+号按钮，添加三个参数框

。

**图1.1.4添加参数**

1. 在第一个参数框的左边框双击输入参数名称Treshold（最小阈值），其右边框双击可以输入整数数值范围为0~255，当您选用不同的阈值算法时，该参数呈现不同的作用。
2. 在第二个参数框的左边框双击输入参数名称MaxValue（最大阈值），其右边框双击可以输入整数数值范围为0~255，当您选用不同的阈值算法时，该参数呈现不同的作用（此参数可不添加，当不添加次参数时，默认最大阈值为255，若添加了一定要设置数值）。
3. 在第三个参数框的左边框双击输入参数名称TresholdType（阈值处理方法），其右边框双击可以输入整数1、2、3、4、5，分别代表二值化阈值处理、反二值化阈值处理、截断阈值化处理、低阈值零处理以及超阈值零处理五种阈值处理方法。

1）二值化阈值处理（cv2.THRESH\_BINARY）：可将灰度值大于最小阈值的设为最大阈值，小于最小阈值的设为 0（最小阈值为Threshold右方输入的值，最大阈值为MaxValue右方输入的值）

2）反二值化阈值处理（cv2.THRESH\_BINARY\_INV）：可将灰度值小于最小阈值的设为最大阈值，大于最小阈值的设为 0（最小阈值为Threshold右方输入的值，最大阈值为MaxValue右方输入的值）

3）截断阈值化处理（cv2.THRESH\_TRUNC）：可将灰度值大于最小阈值的设置为最小阈值，其他的保持不变（最小阈值为Threshold右方输入的值。）

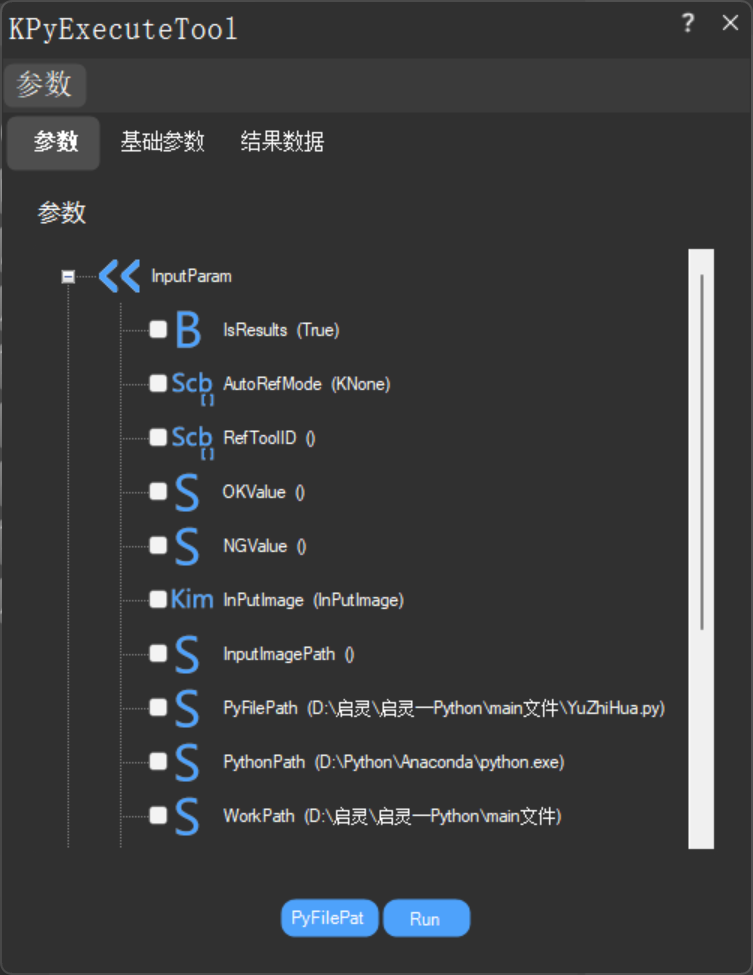
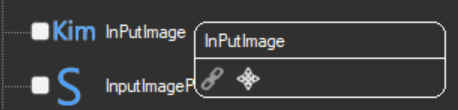
4）低阈值零处理（cv2.THRESH\_TOZERO）：可将灰度值小于最小阈值的设置为0，其他的保持不变（最小阈值为Threshold右方输入的值。）

5）超阈值零处理（cv2.THRESH\_TOZERO\_INV）：可将灰度值大于最小阈值的设置为0，其他的保持不变（最小阈值为Threshold右方输入的值。）

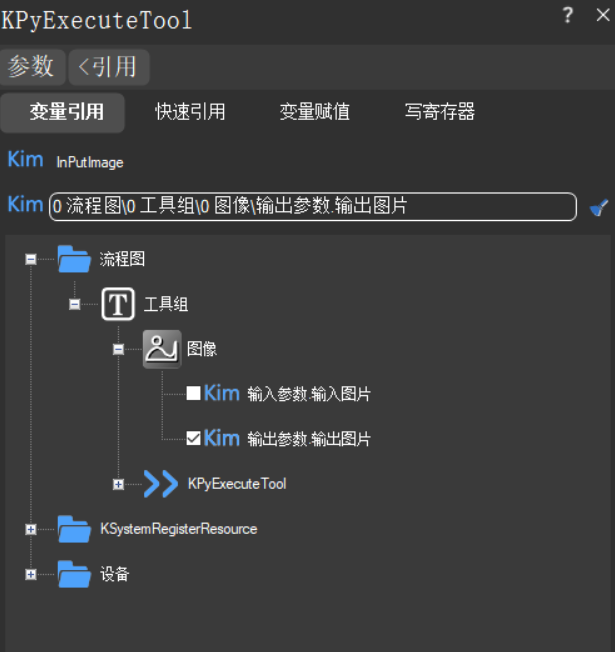
1.1.2参数设置

输入图像的引用：

(1)点击InputParam左边白方块中展开输入参数，点击InPutImage右方括号中的InPutImage进行展开，然后点击引用图标进入输入图像引用界面。如图1.1.5所示。



**图1.1.5 输入图像引用**

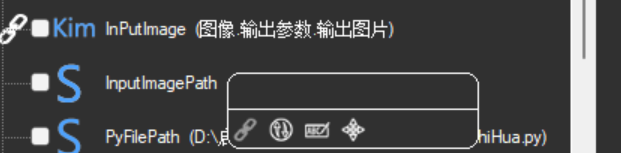
(2)进入引用界面之后，依次展开文件找到自己需要引用 的图像勾选进行引用。如图1.1.6所示。

**图1.1.6 输入图像引用**

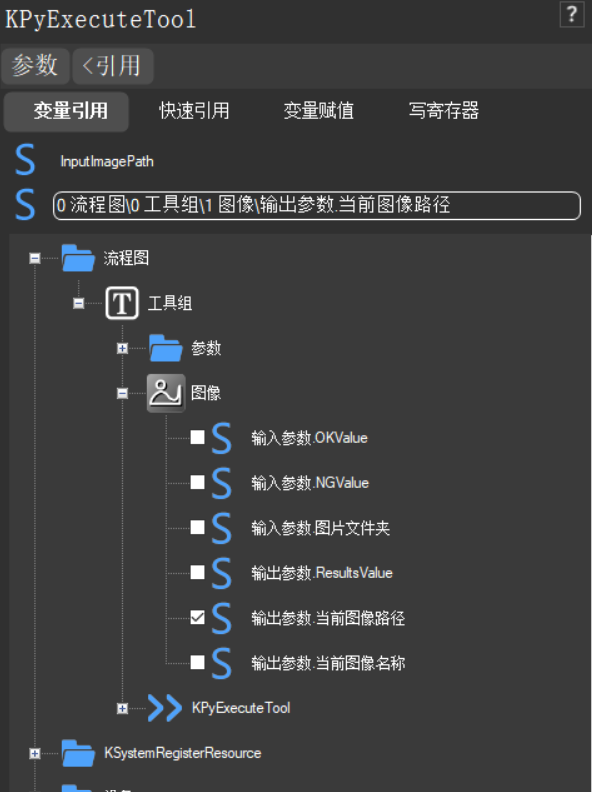
输入图像路径的引用：

(1)点击InputParam左边白方块展开输入参数，点击InPutImagePath右方括号展开，然后点击引用图标进入输入图像引用界面。如图1.1.7所示。





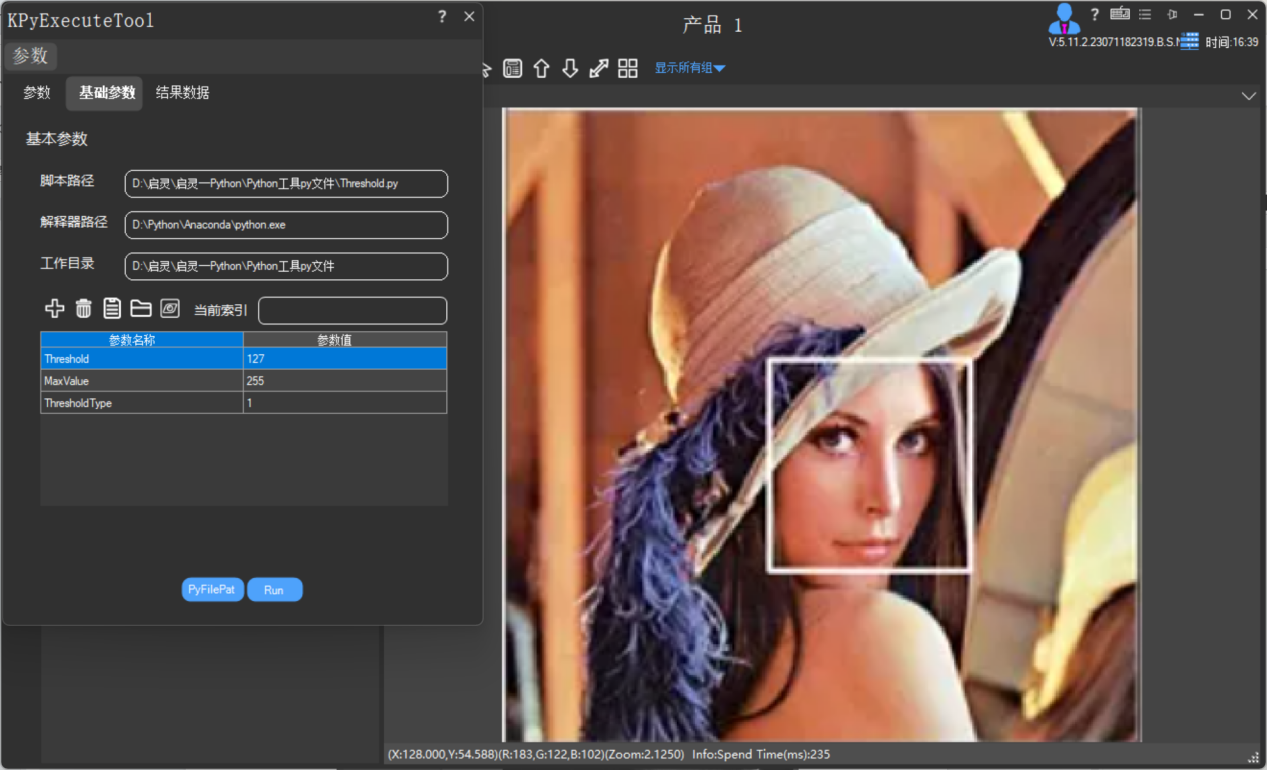
**图1.1.7 输入图像路径的引用**

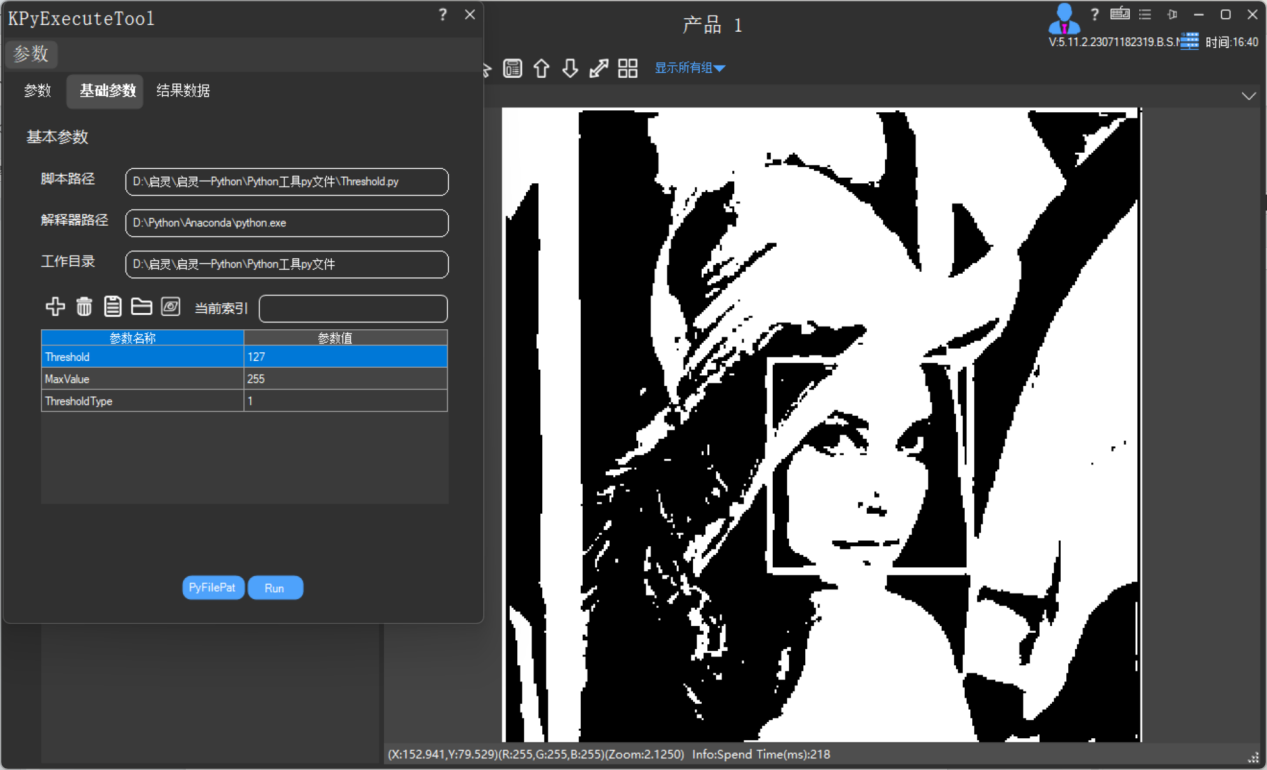
 (2)进入引用界面之后，依次展开文件找到自己需要引用的图像路径勾选进行引用。如图1.1.8所示。

**图1.1.8 输入图像路径的引用**

1.1.3实例

回到基础参数页面，将Threshold的参数值设置为127，MaxValue的参数值设置为255，ThresholdType的参数值设置为1，点击Run按钮，原图和效果图如下图所示。





**图1.1.9 原图及效果图**