

版本: Visual Studio 2022

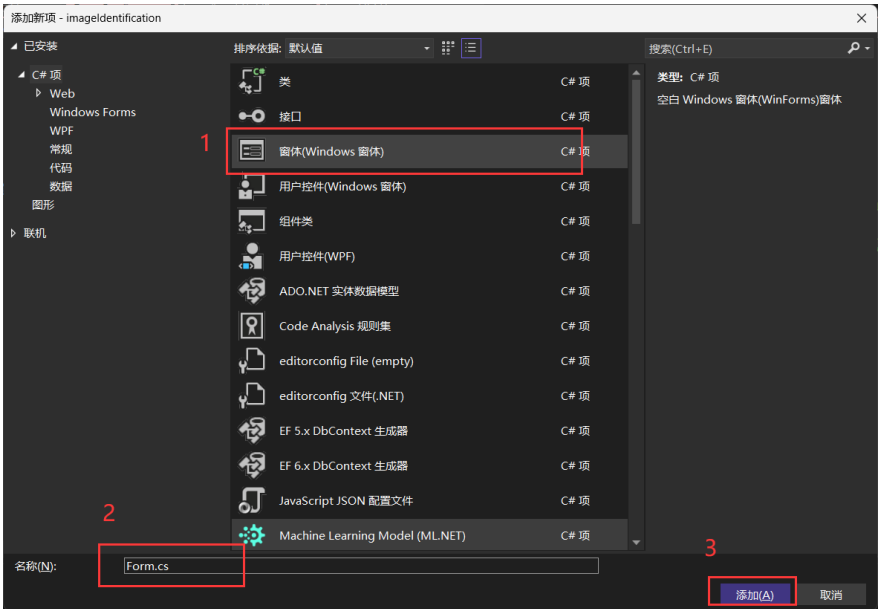
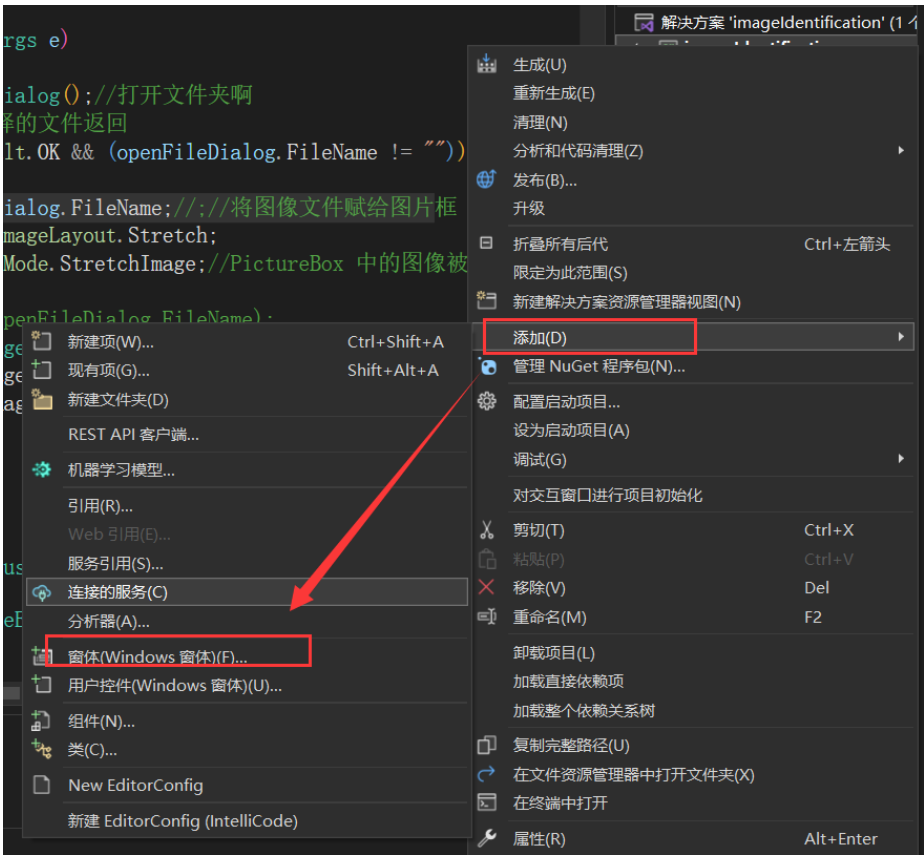
语言: C#

时间: 2024. 4. 7

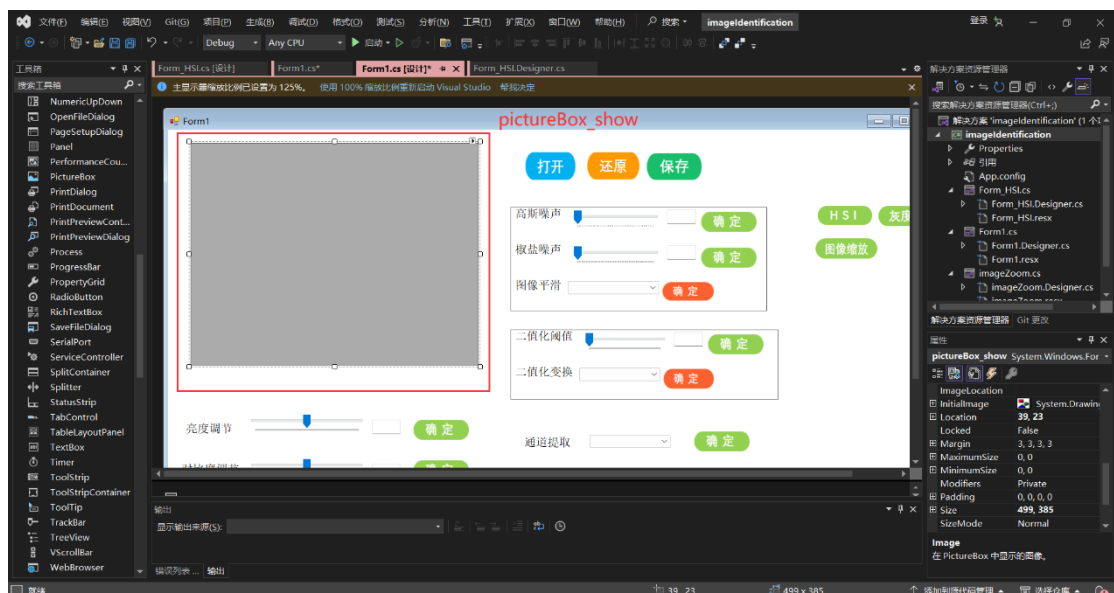
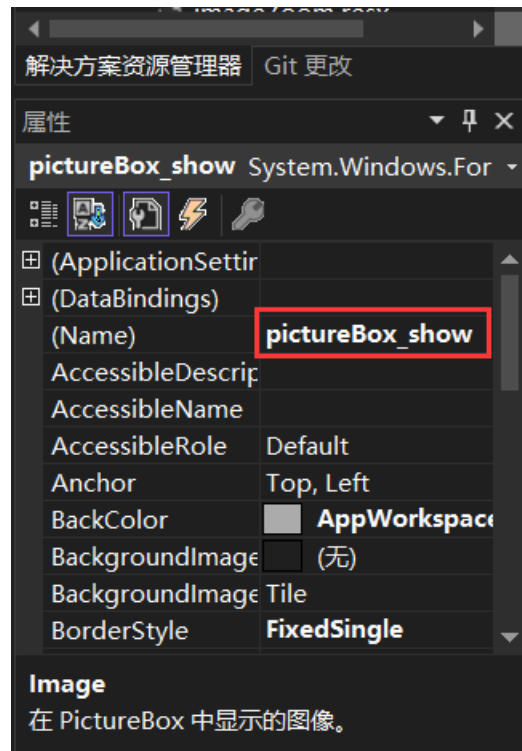
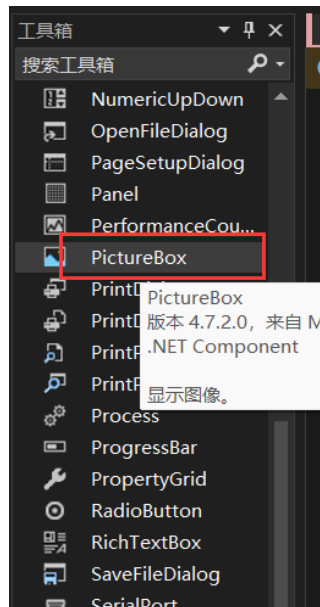
功能: 建立一个图像处理程序框架，实现图像的读入和存储

步骤:

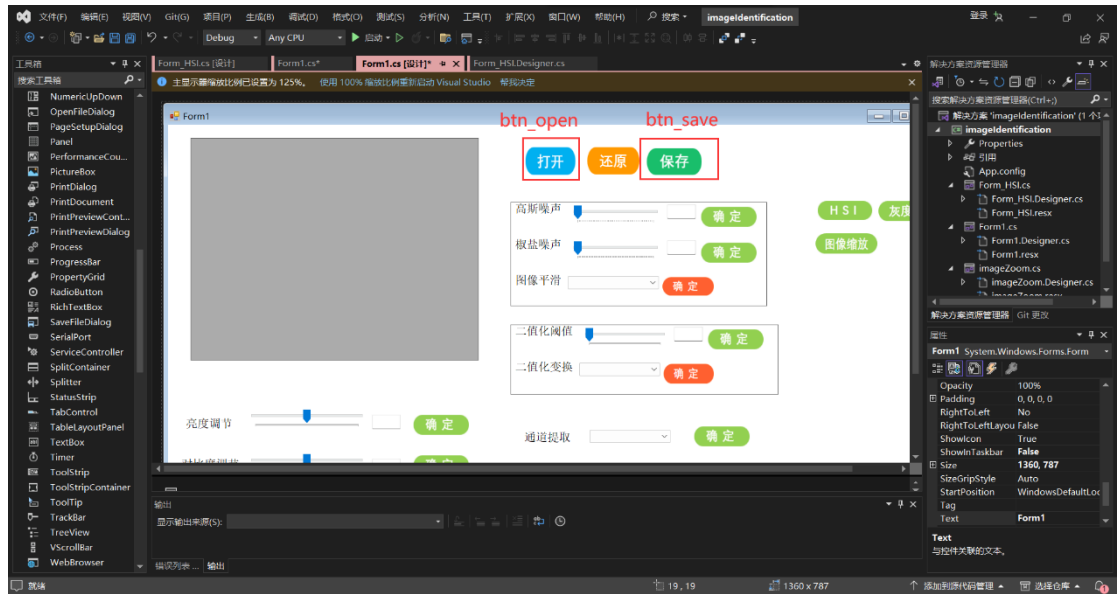
1、新建一个窗体，取名为 Form.cs



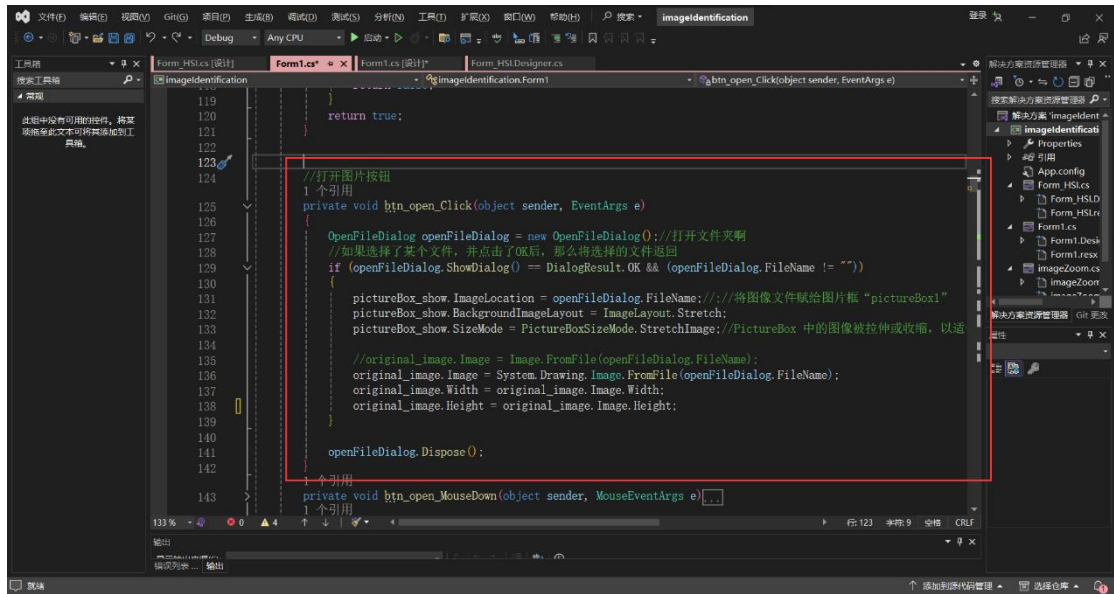
2、点击 Form.cs，在右侧的工具栏中选择 PictureBox 组件，并拖拽到窗体的相关位置，点击窗体上的 PictureBox，在右下角的属性表中修改其名字为 pictureBox\_show，表示其为图片展示窗口，其余相关属性可根据需要进行设置



3、选择左侧工具栏中的 Button 组件，添加两个按钮在窗体中，一个按钮取名为 btn\_open，表示为打开图片按钮，并将其 text 属性设置为“打开”，另一个按钮取名为 btn\_save，表示为保存图片按钮，并将其 text 属性设置为“保存”，随后根据需要可以自行设置其余属性，这里我为了区别，将两个按钮添加了不同的背景图片



4、双击 btn\_open 按钮，跳转至 Form.cs 代码撰写窗口，且光标自动跳转到了 btn\_open\_Click 函数中，该函数即为 btn\_open 按钮的点击事件，在该函数中我们需要实现的功能为：当点击 btn\_open 按钮时，弹出文件夹，在文件夹中我们可以选择图片，并将其显示在 pictureBox\_show 组件之中。



代码解释：这段代码是一个用于打开图片并显示在 PictureBox 组件中的事件处理方法。它是一个按钮的点击事件处理方法，当用户点击按钮时，会触发该方法。

首先，创建一个 OpenFileDialog 对象，用于选择要打开的图像文件。

如果用户选择了一个文件并点击了“OK”按钮，则进入 if 语句块。

在 if 语句块中，将选中的图像文件路径赋值给 pictureBox\_show 控件的 ImageLocation 属性，这样图像会显示在 PictureBox 中。

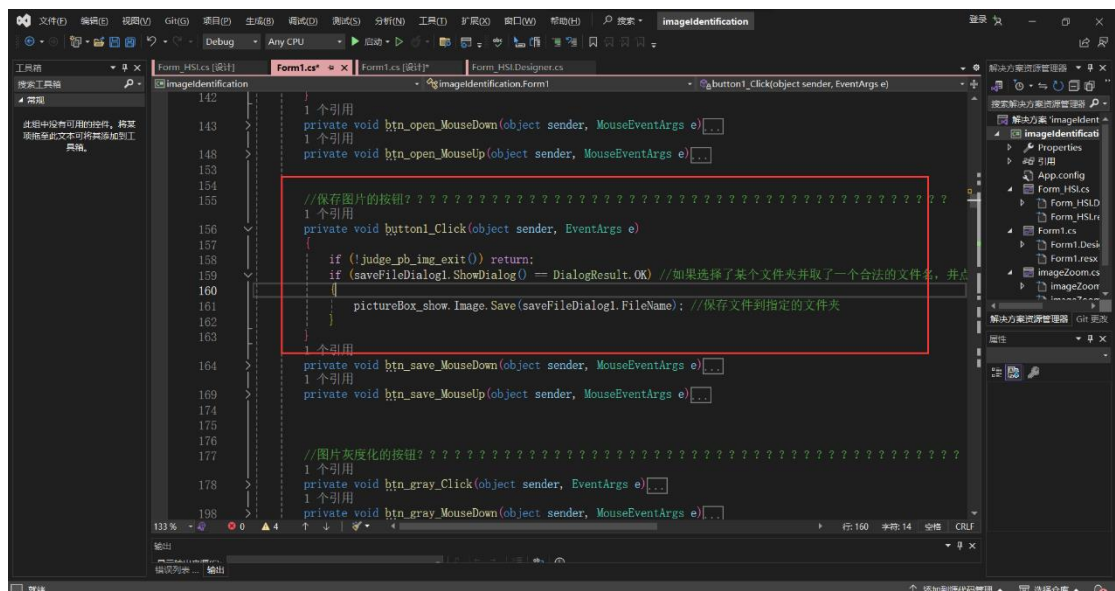
设置 pictureBox\_show 控件的图像布局方式为拉伸(ImageLayout.Stretch)，这样图像将会充满整个 PictureBox 控件。

设置 pictureBox\_show 控件的图像显示模式为拉伸适应 PictureBox 的大小(PictureBoxSizeMode.StretchImage)。

将图像文件加载到 original\_image 控件中，并设置 original\_image 控件的宽度和高度与图像的宽度和高度相等。注意，这行代码可能是重复的，可以根据需要删除。

最后，释放打开文件对话框对象的资源。

## 5、同理编写图片保存按钮的点击事件：



代码解释：

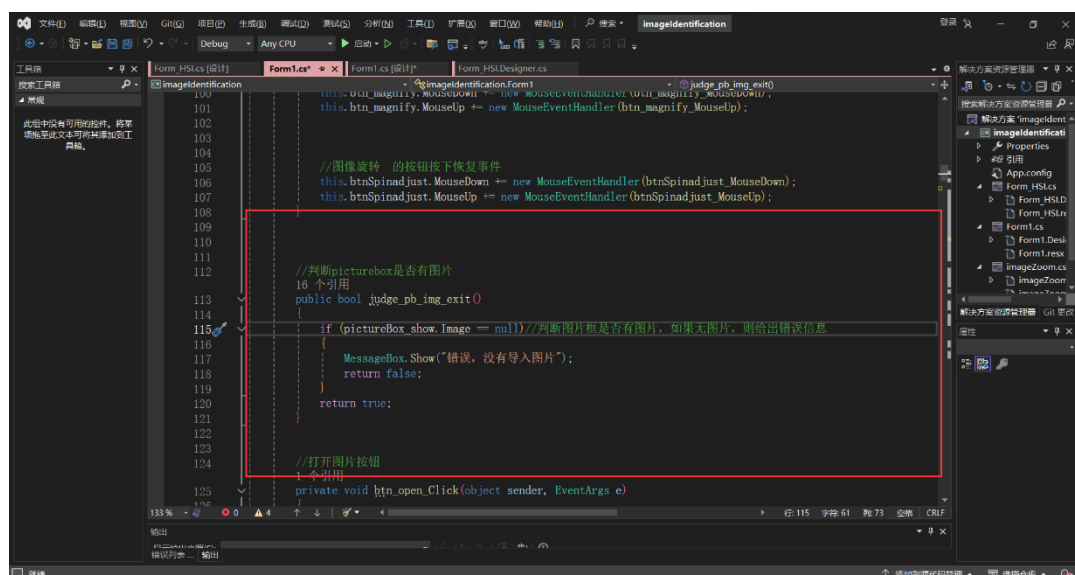
首先，调用 judge\_pb\_img\_exit() 方法，该方法用于判断是否存在要保存的图片。如果不存在，通过 return 语句提前退出方法，不执行后续的保存操作。

然后，打开一个保存文件对话框（saveFileDialog1），供用户选择保存图像的文件夹和文件名。

如果用户选择了一个文件夹并取了一个合法的文件名，并点击了“OK”按钮，则进入 if 语句块。

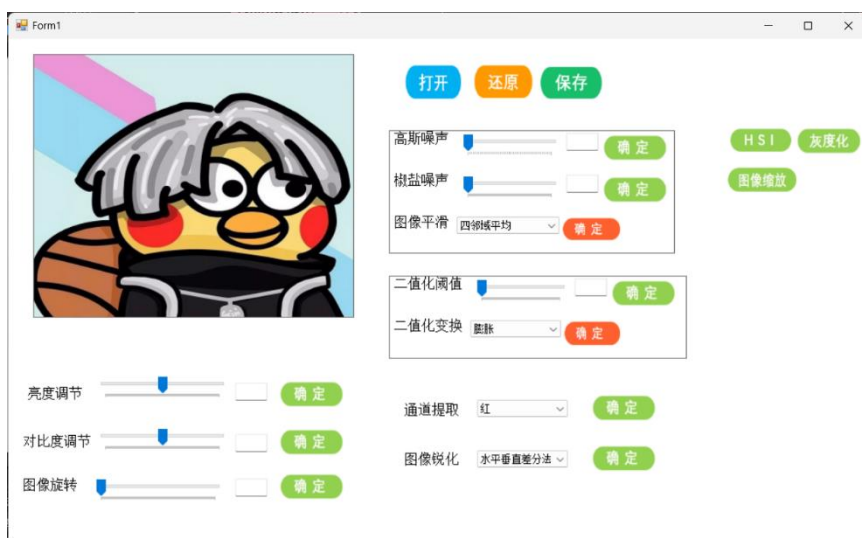
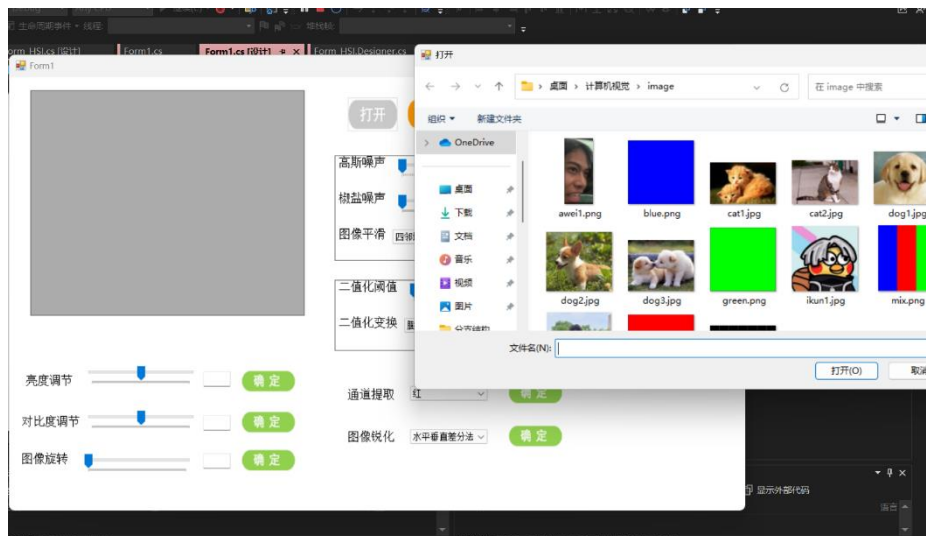
在 if 语句块中，使用 pictureBox\_show 控件的 Image.Save 方法将图像保存到指定的文件夹中，保存的文件路径由 saveFileDialog1.FileName 提供。

6、为了保证代码的严谨性和健壮性，我们需要在保存图片或者后续一系列图像处理的过程中，首先判断 pictureBox\_show 组件中是否有图片，如若没有图片，会弹窗提示用户，防止爆 null。因此我编写了一个名为 judge\_pb\_img\_exit 的函数，用于判断 pictureBox 中是否有图片，若没有图片，则弹窗提示用户，并给出错误信息：

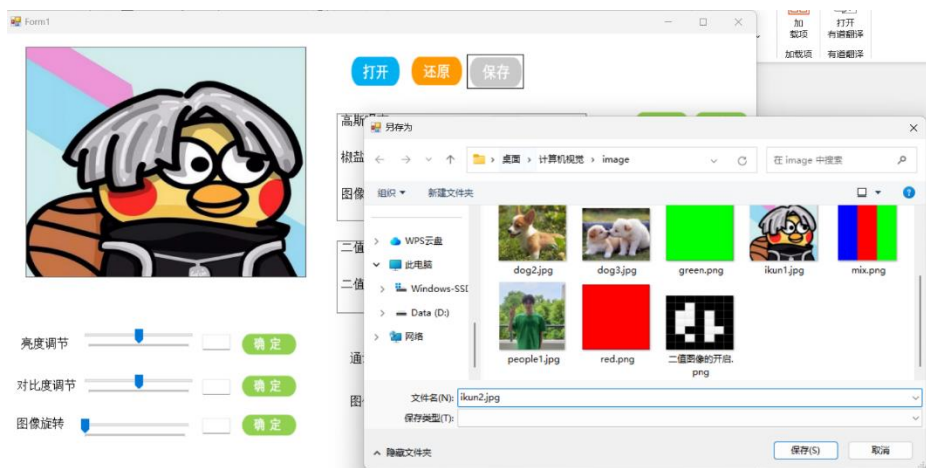


7、演示效果：

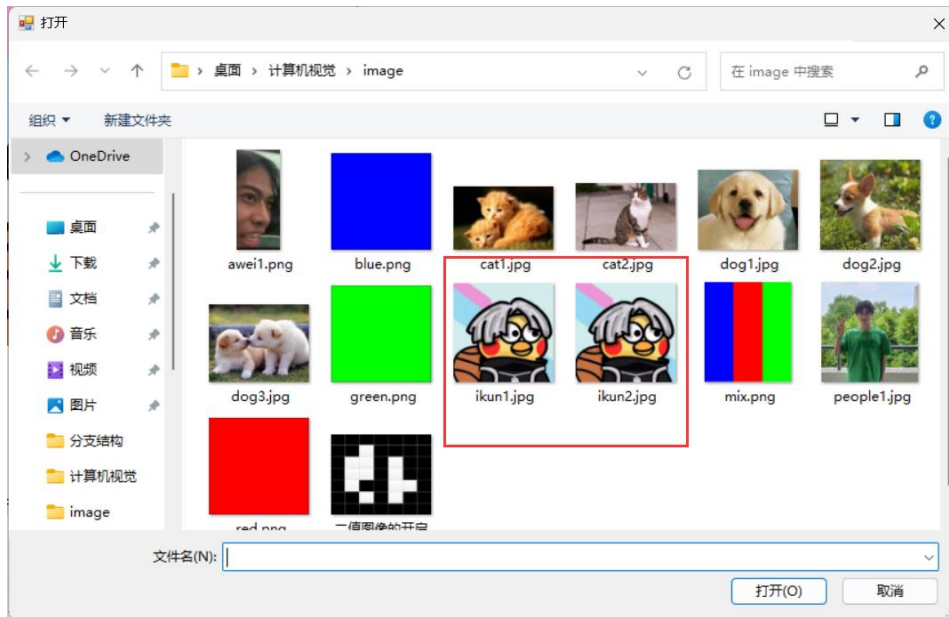
打开图片：



保存图片：







8、至此，利用 C#处理图像的前期准备工作结束，后续将根据需要添加更多的功能。