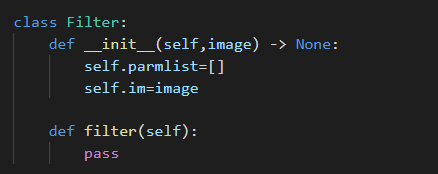
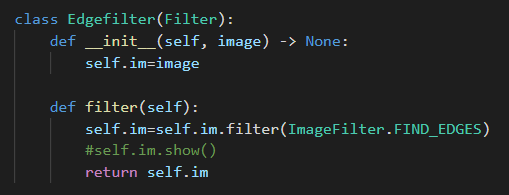
1. 基类Filter

**Image为Image的实例，parmlist为可能的列表**



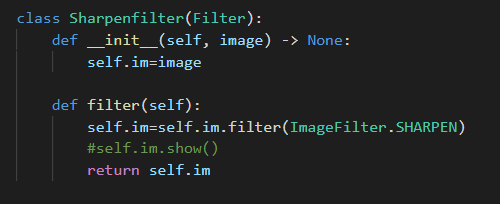
1. Filter的子类
2. Edgefilter

**用于对图片进行边缘提取**



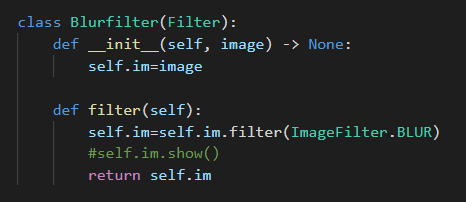
1. Sharpenfilter

**对图片进行锐化**



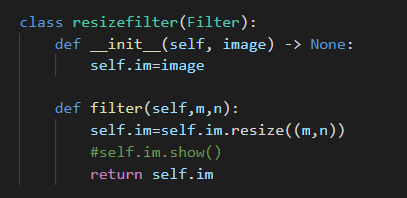
1. Blurfilter

**对图片进行模糊处理**



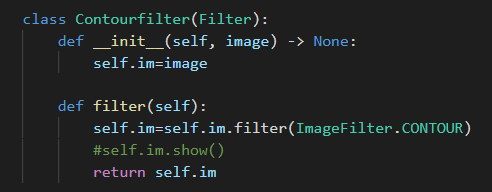
1. Resizefilter

**调整图片的尺寸**



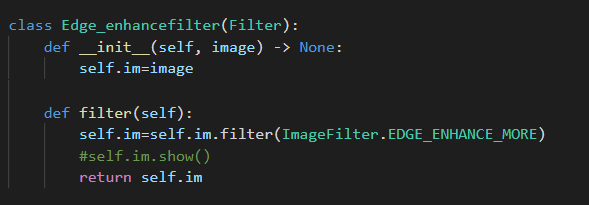
1. Contourfilter

**提取图片的浮雕**



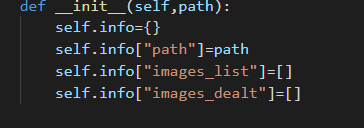
1. Edge\_enhancefilter

**增强图片的边缘**



1. ImageShop
2. 初始化

**利用字典来储存属性，分别为图片文件（目录）的路径，Image实例的列表，处理后的图片列表和图片的类型**



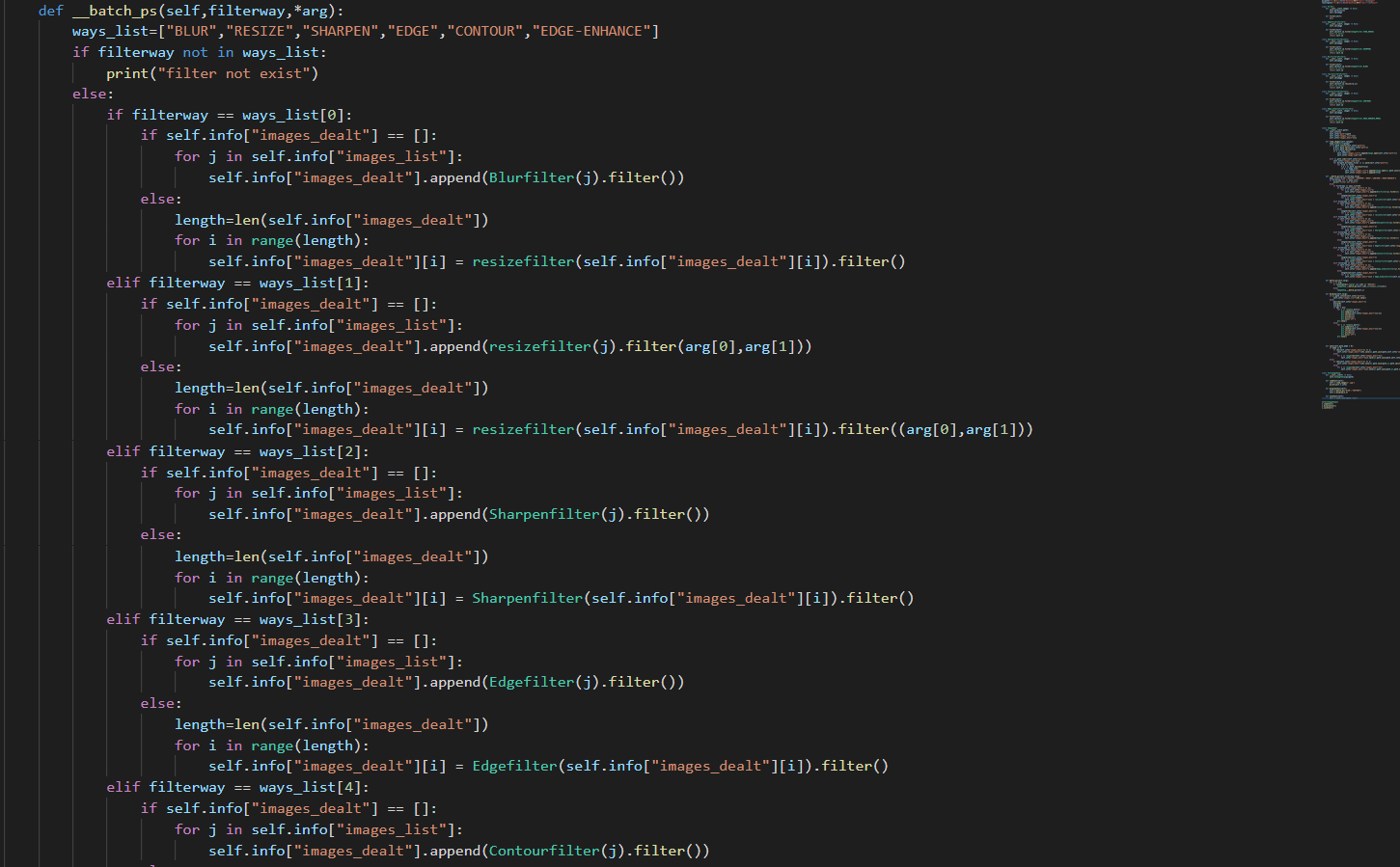
1. load\_images

**允许输入一个或者多个要加载的图片类型；同时可以判断是图片文件还是目录，如果是文件那么图片类型是单个字符串，如果是目录将会对该目录下的所有符合格式要求的图片进行加载，用列表储存实例和图片的类型。**



1. 内部方法\_\_batch\_ps

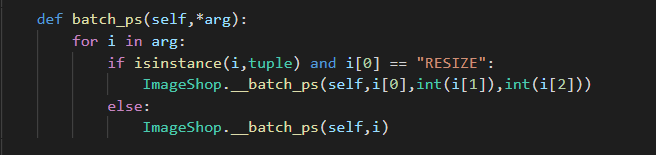
**首先给出对应Filter子类的过滤器的操作列表，根据用户的输入执行对应的操作，对实例列表中的图片进行批量处理；考虑到可能存在同时有多个操作，此时要判断图片是否已经经过操作（如果处理后的图片列表为空则图片没有进行过操作，反之则经过），如果经过操作那么要对储存处理后的图片列表进行修改，反之对原始列表进行修改**





1. 外部方法batch\_ps

**用户可以输入一个或者多个操作指令，如果操作指令有参数，要按照（指令，参数）的形式输入**



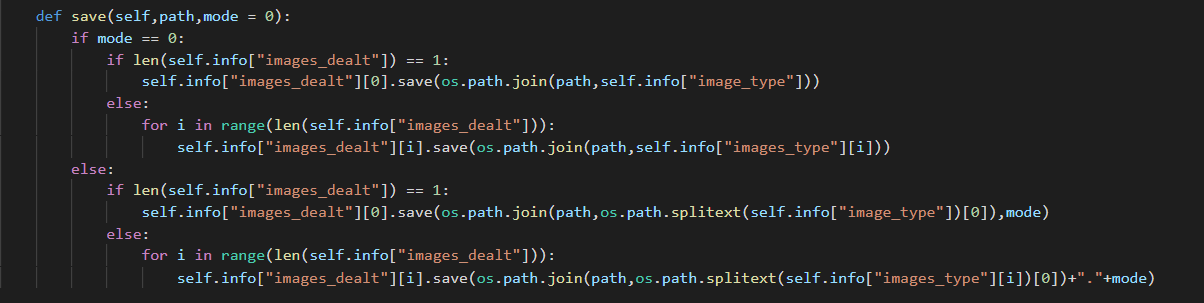
1. Display

**对处理结果进行展示，如果为多个图片，用户可以输入想要的行数和列数，同时能同时显示的最大数量为图片总数和行列数的乘积两者的最小值；**



1. Save

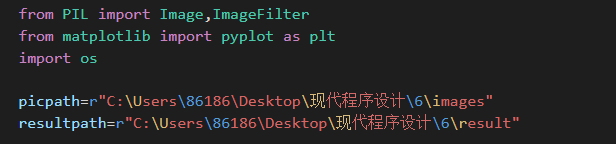
**可以根据用户要求对图片进行保存，mode参数控制保存的格式，默认为0，即保留原来的格式；同时也可以输入想要保存的格式，如“png”，“gif”等**

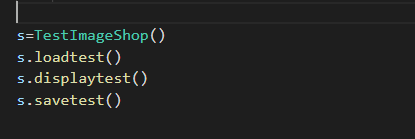


1. 测试类TestImageShop

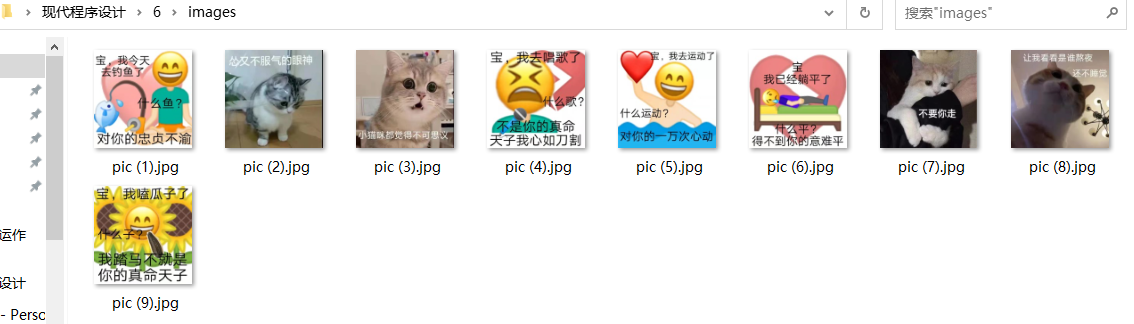
**分别对图片的加载功能（选择加载jpg格式的图片），批量处理功能（同时对所有图片进行模糊并且调整大小为300\*500），处理结果展示（以3\*3的形式展示），保存处理结果（将图片保存为png格式）进行测试**





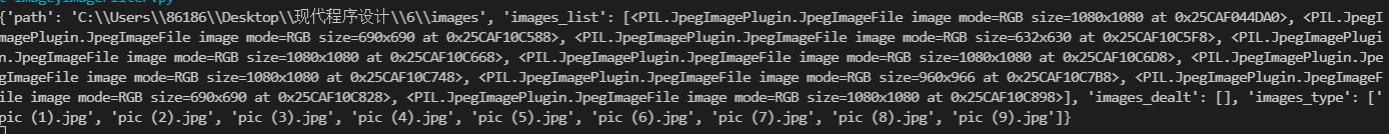


原始图片：

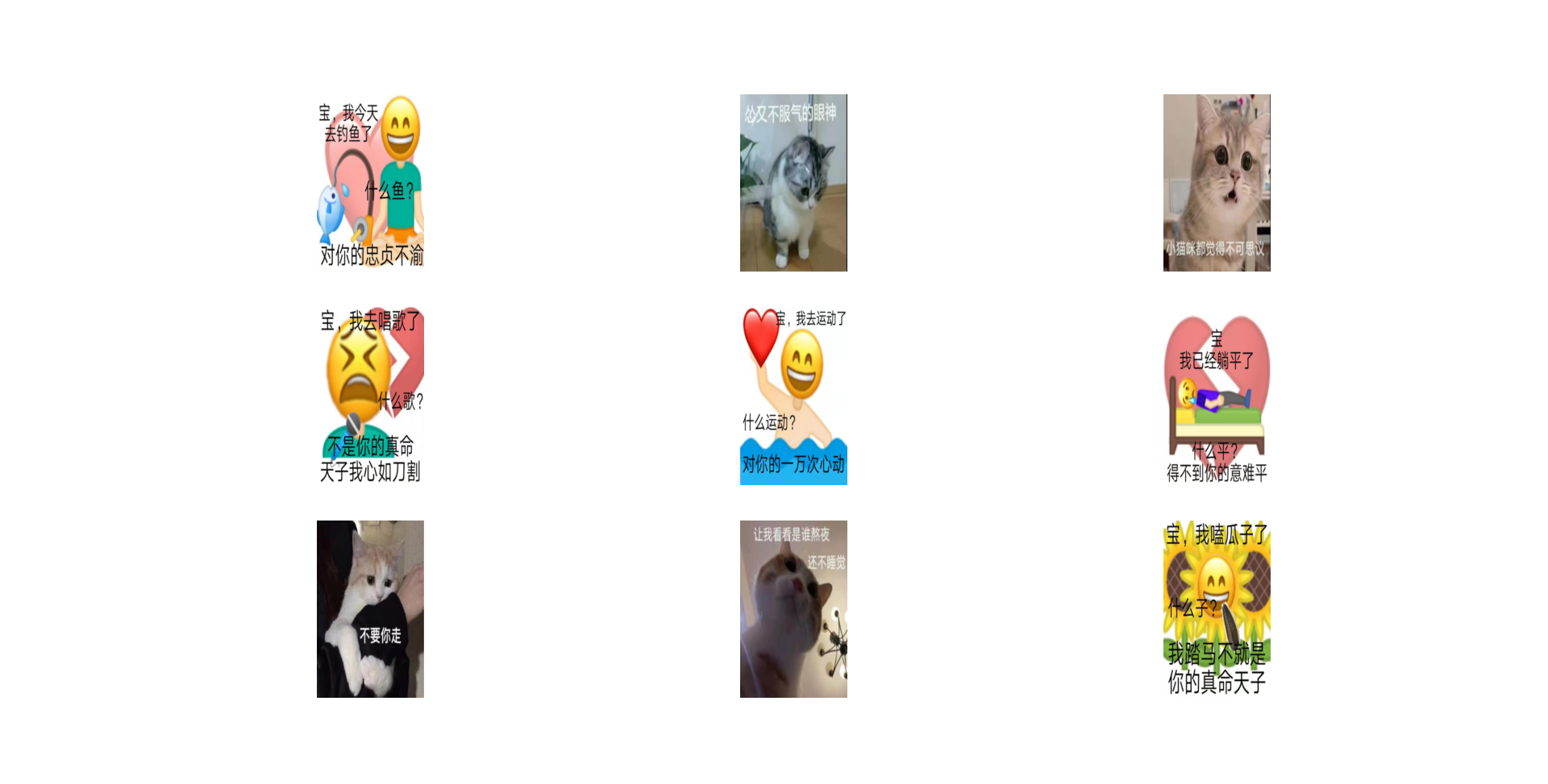


运行结果：

1. 图片加载的字典



1. 处理后的效果展示，可以发现比列发生了变化



1. 保存的图片结果

