



中国传媒大学
CommandButto
COMMUNICATION UNIVERSITY OF CHINA

题目：基于 GUI 的图像处理系统

学年学期：大一下学期

课程名称：Matlab 语言程序设计

课程编号：2031030203

课程序号：01

任课教师：孟明

姓 名：吴宇晴

学 号：202213103032

评分区域（由阅卷老师填写）：

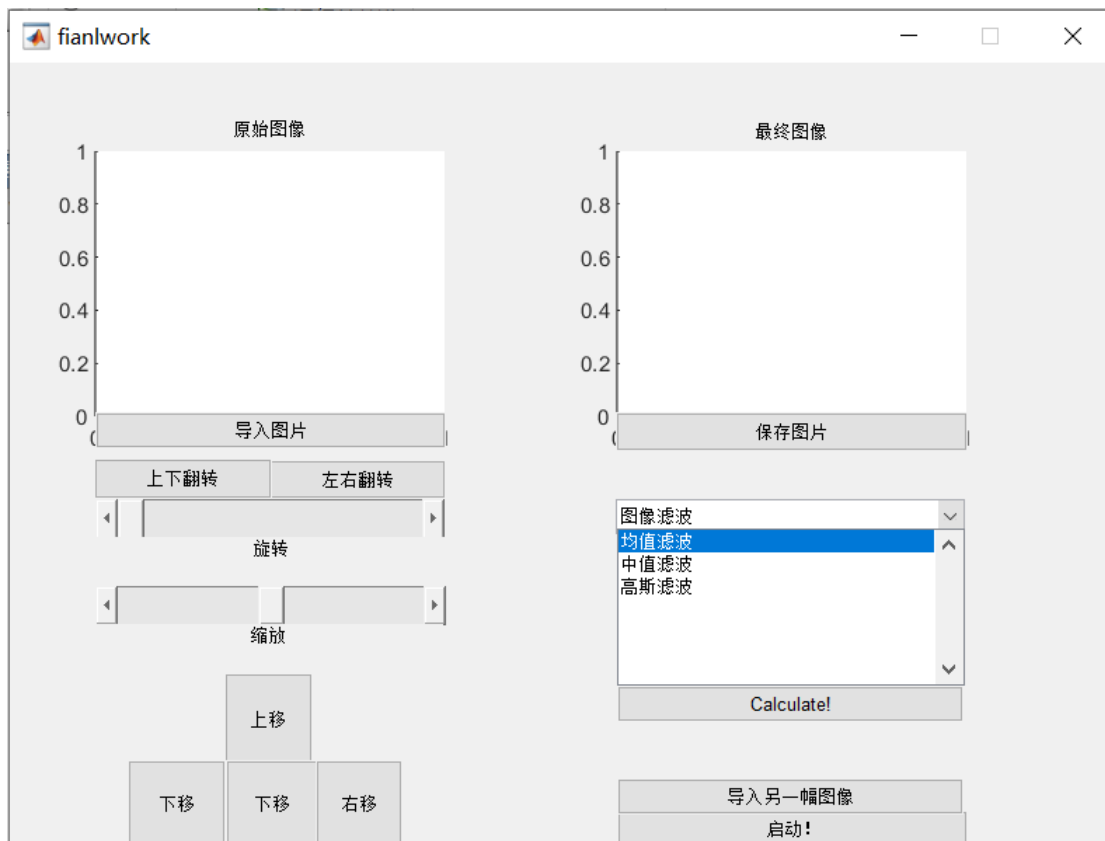
结课成绩：

总评成绩：

提交时间： 2023 年 6 月 29 日

一、介绍：

选取了题目二完成。要求的功能全部实现，并设计了较为简洁易懂的 GUI 界面。下面分功能模块介绍各个功能。



1. 导入图片

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename, pathname] = uigetfile({'*.jpg;*.png;*.bmp;*.tif'}, '选择图像文件');
raw= imread(fullfile(pathname, filename));
axes(handles.axes1);
imshow(raw);
axes(handles.axes2);
imshow(raw);%创建对话框导入图像并展示在两个axes中
```

用 `uigetfile` 创建对话框，让用户选择要导入的图片，并记录其地址，读取图片后用 `imshow` 展示在画框 1 和画框 2 中。

2. 上下翻转\左右翻转

```
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
axes(handles.axes1);
pic=getimage(handles.axes2);
pic=flip(pic,1);
axes(handles.axes2);
imshow(pic);%上下翻转模块的实现
```

将画框中展示的图片读取，然后用 flip 函数对其进行上下\左右的镜像操作。
完成后用
Imshow 展示。

3. 旋转

```
% --- Executes on slider movement.
function slider1_Callback(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider
val=get(hObject,'value');
angle=val*360;
sub=getimage(handles.axes1);
outcome=imrotate(sub,angle,'bilinear');
imshow(outcome);
```

用 get 函数读取滑块的值，并将其乘以 360，代表 360° 读取要处理的图像后用 imrotate 对图片进行旋转，值得注意的是 imrotate 是在读取的图片的基础上再进行操作，所以每次需要读取原图片，才能保证每次的度数不与上一次的叠加。

4. 缩放

```
% Hints: get(hObject,'Value') returns position of slider
%         get(hObject,'Min') and get(hObject,'Max') to determine range of slider
val=get(hObject,'value');
if val>0.5
    muti=val*4-1;
else
    muti=val*2;
end%缩小倍率0~1; 放大倍率1~3
sub=getimage(handles.axes1);
outcome=imresize(sub,muti,'bilinear');
save=gca;
xlim=get(save,'xlim');
ylim=get(save,'ylim');
imshow(outcome);
set(save,'xlim',xlim);
set(save,'ylim',ylim);%使x,y轴与图片的比例保持不变,避免出现像素点变化而图片大小没变的情况
```

用 get 读取滑块的值，并对这个值进行分段， <0.5 的部分 $\times 2$ 完成 $0\sim 1$ 的缩小， >0.5 的部分 $\times 4-1$ 完成 $1\sim 3$ 倍的放大。用 imresize 函数实现图片的缩放，并保存坐标轴此时的状态，在展示图片后还原，这样才能避免坐标轴的比例与图片一起缩放。

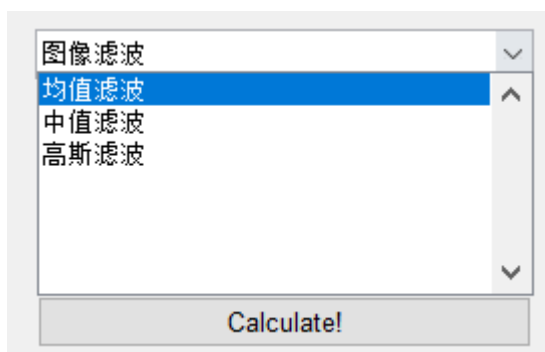
5. 上\下\左\右平移

```
% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
deltax=-10;deltay=0;
T=maketform('affine',[1 0 0;0 1 0;deltax deltay 1]);
sub=getimage(handles.axes2);
outcome=imtransform(sub,T,'XData',[1 size(sub,2)],'YData',[1,size(sub,1)],'FillValue',255);
imshow(outcome);
```

用 transform 对图片的位置进行调整，还存在一定的问题，移出视野的图片部分会被裁剪掉，移回来时无法显示了。

6. 滤波

这边的高级功能做了一个下拉式菜单，选择对应的高级功能可以继续进行选择具体的处理这大大节约了空间。



具体就是读取菜单的选择的值，用 switch 函数将不同的选择与不同的处理模块相连

```
switch sig
    case 1
        switch tar
            case 1|
                H = fspecial('average',[3,3]);
                outcome = imfilter(raw,H);
                axes(handles.axes2);
                imshow(outcome);
            case 2
                outcome(:,:,1)=medfilt2(raw(:,:,1),[3 3]);
                outcome(:,:,2)=medfilt2(raw(:,:,2),[3 3]);
                outcome(:,:,3)=medfilt2(raw(:,:,3),[3 3]);
                axes(handles.axes2);
                imshow(outcome);
            case 3
                W = fspecial('gaussian',[3,3],1);
                outcome(:,:,1)=imfilter(raw(:,:,1), W, 'replicate');
                outcome(:,:,2)=imfilter(raw(:,:,2), W, 'replicate');
                outcome(:,:,3)=imfilter(raw(:,:,3), W, 'replicate');
                axes(handles.axes2);
                imshow(outcome);
```

均值滤波运用了 matlab 自带的函数，先用 fspecial 制作一个过滤器，再用 imfilter 对图像进行过滤。

中值滤波采用 medfilt2 函数进行处理，分别对图像的红，绿，蓝三层进行处理。高斯滤波也是用 fspecial 制作的过滤器，再分别对图像的每个颜色层进行的处理。

7. 图像增强

```
case 2
    switch tar
    case 1
        outcome=rgb2gray(raw);
        axes(handles.axes2);
        imshow(outcome);
    case 2
        mid=rgb2gray(raw);
        outcome=histeq(mid);
        axes(handles.axes2);
        imshow(outcome);
    case 3
        mid=rgb2gray(raw);
        outcome=imadjust(mid);
        axes(handles.axes2);
        imshow(outcome);
    end
```

灰度转化是使用的自带的 `rgb2gray` 进行的转换，直方图均衡化则是先将图片转化为黑白图片，再用 `histeq` 进行处理，在学习过程中也了解到有许多其他的方法实现这个功能，效果也会更好。

对比度增强的实现是先将图片转化为灰色，再使用 `imadjust` 实现增强，缺点是没有将图片转化回彩色图片。

8. 图像分割

```

case 3
    switch tar
    case 1
        mid(:,:,1)=medfilt2(raw(:,:,1),[3 3]);
        mid(:,:,2)=medfilt2(raw(:,:,2),[3 3]);
        mid(:,:,3)=medfilt2(raw(:,:,3),[3 3]);
        %高斯滤波，去除噪点
        mid=rgb2gray(mid);
        outcome=imbinarize(mid);
        axes(handles.axes2);
        imshow(outcome);
    case 2
        mid(:,:,1)=medfilt2(raw(:,:,1),[3 3]);
        mid(:,:,2)=medfilt2(raw(:,:,2),[3 3]);
        mid(:,:,3)=medfilt2(raw(:,:,3),[3 3]);
        %高斯滤波，去除噪点
        mid=rgb2gray(mid);
        outcome=edge(mid,'canny');
        axes(handles.axes2);
        imshow(outcome);
    end

```

阈值分割，为了实现更好的效果，先用高斯滤波去除图像的噪点，再用 `rgb2gray` 函数将图片进行转换，然后使用 matlab 自带的函数 `imbinarize` 完成对图片的阈值分割。边缘检测也是类似，先用高斯滤波去除图像的噪点，再用 `rgb2gray` 函数将图片进行转换，然后用 `edge` 函数实现边缘检测，使用 `canny` 算子。

9. 图像融合

```

% --- Executes on button press in pushbutton12.
function pushbutton12_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename2, pathname2] = uigetfile({'*.jpg;*.png;*.bmp;*.tif'}, '选择图像文件');
pic1=getimage(handles.axes1);
pic2= imread(fullfile(pathname2, filename2));
outcome = imfuse(pic1,pic2,'blend','Scaling','joint');
axes(handles.axes2);
imshow(outcome);

```

先导入需要融合的另一张图片，读取两张图片的信息后，用 `imfuse` 函数将两张图片进行融合，不过问题是融合比较固定，还不能够高程度的自定义。

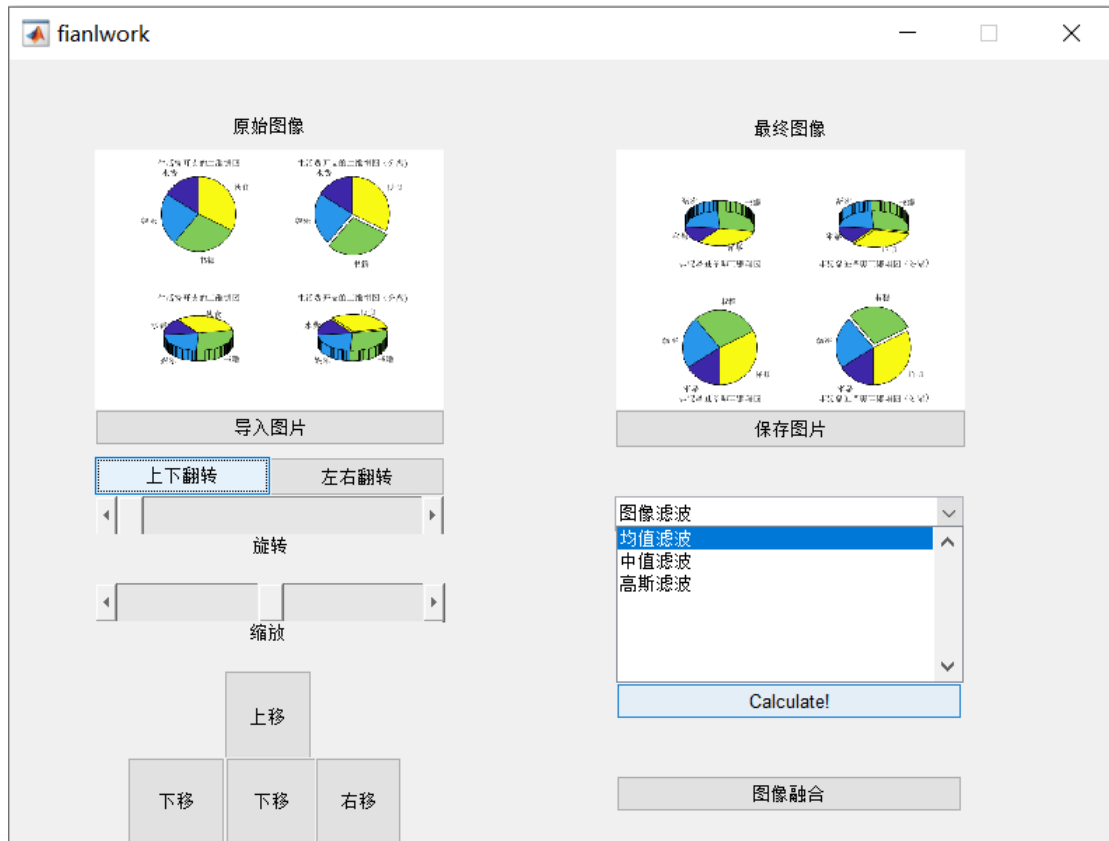
10. 保存

```
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename, pathname] = uiputfile('*.jpg','保存图片到');
I=getimage(handles.axes2);
add=fullfile(pathname, filename);
imwrite(I,add,'jpg');
```

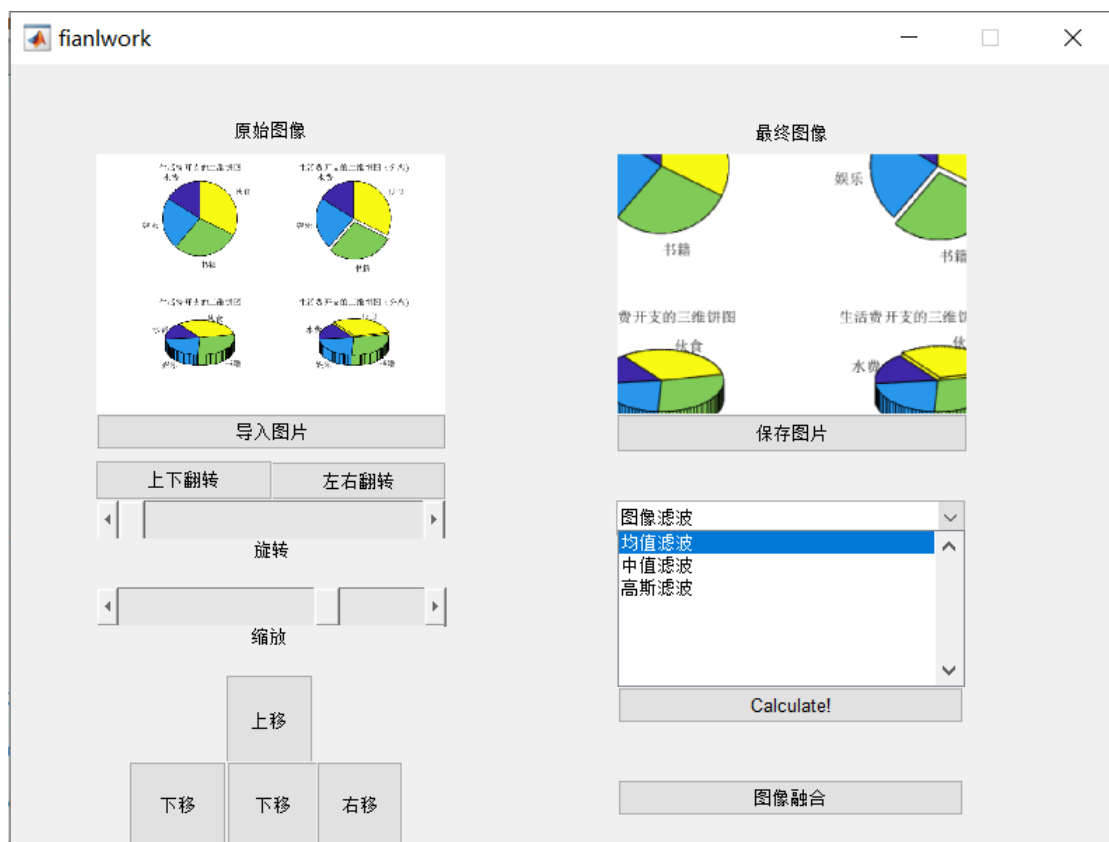
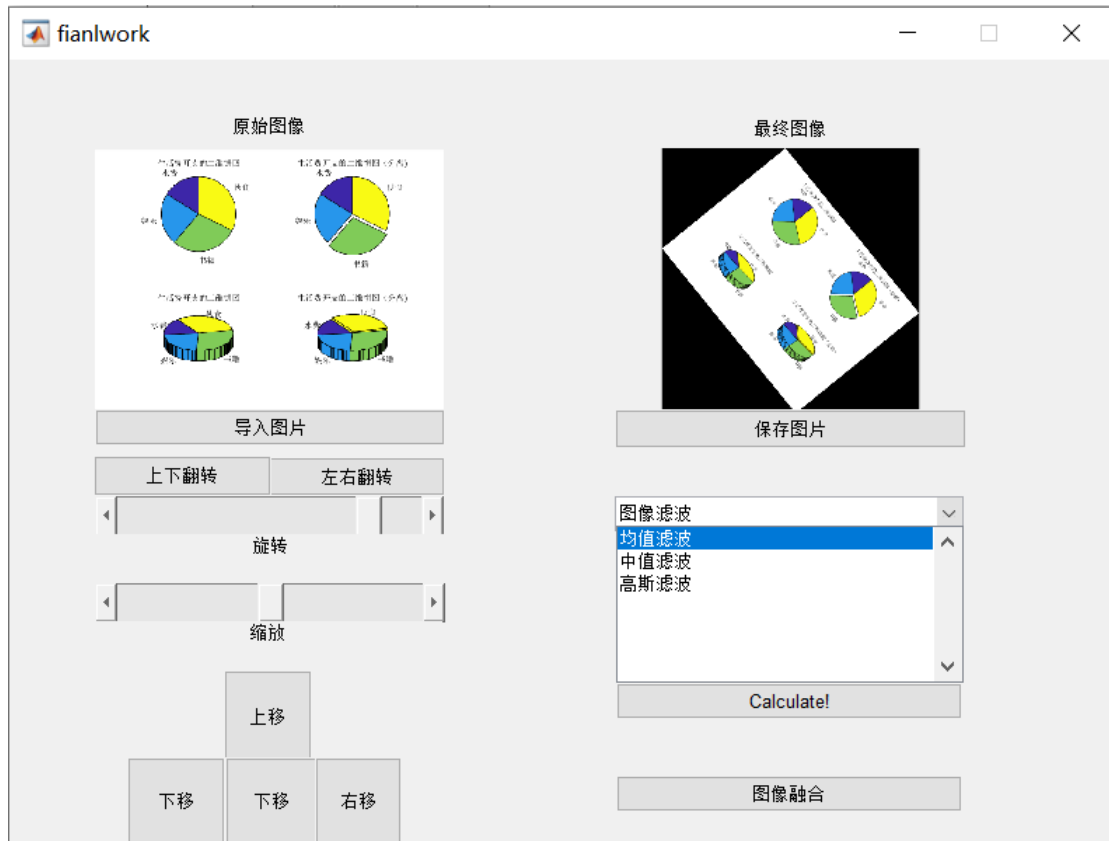
用 uiputfile 建立会话框，选择要保存的地址，记录该地址，再用 imwrite 将图像保存到该地址。

二、程序运行情况

1. 上下\左右旋转

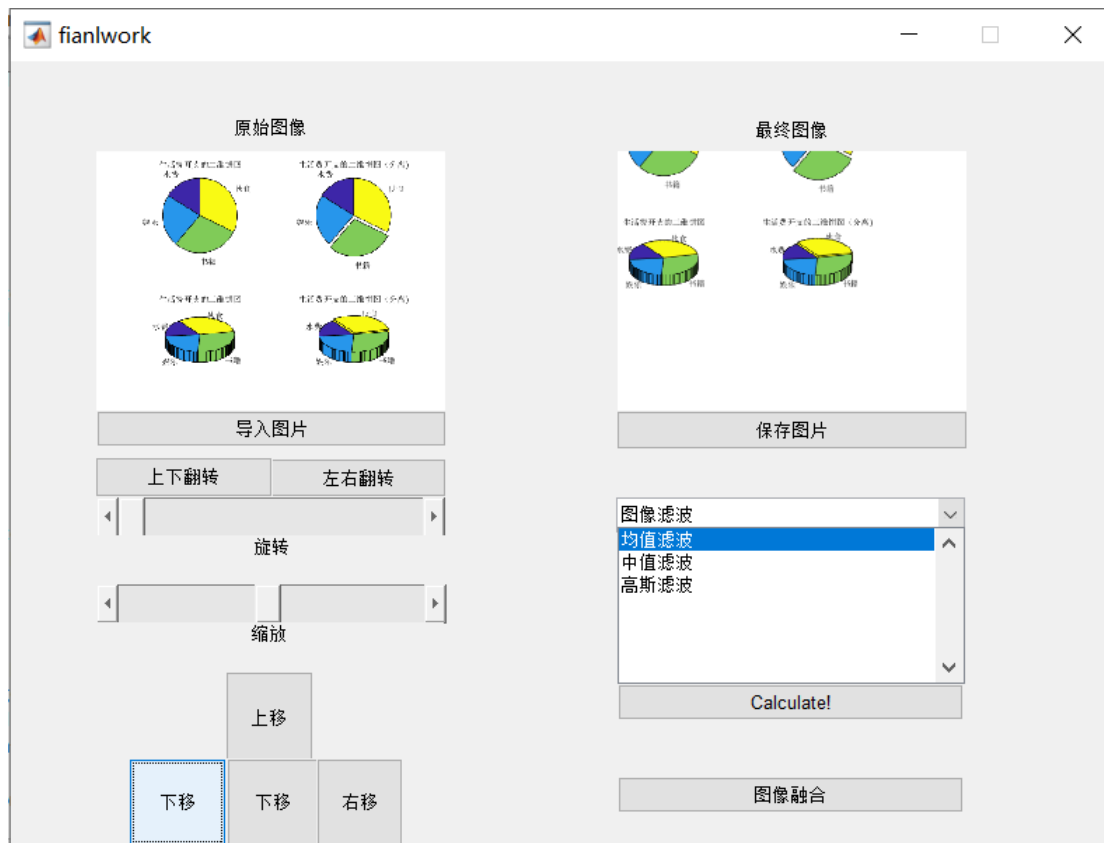


2. 旋转、缩放





3. 上\下\左\右平移

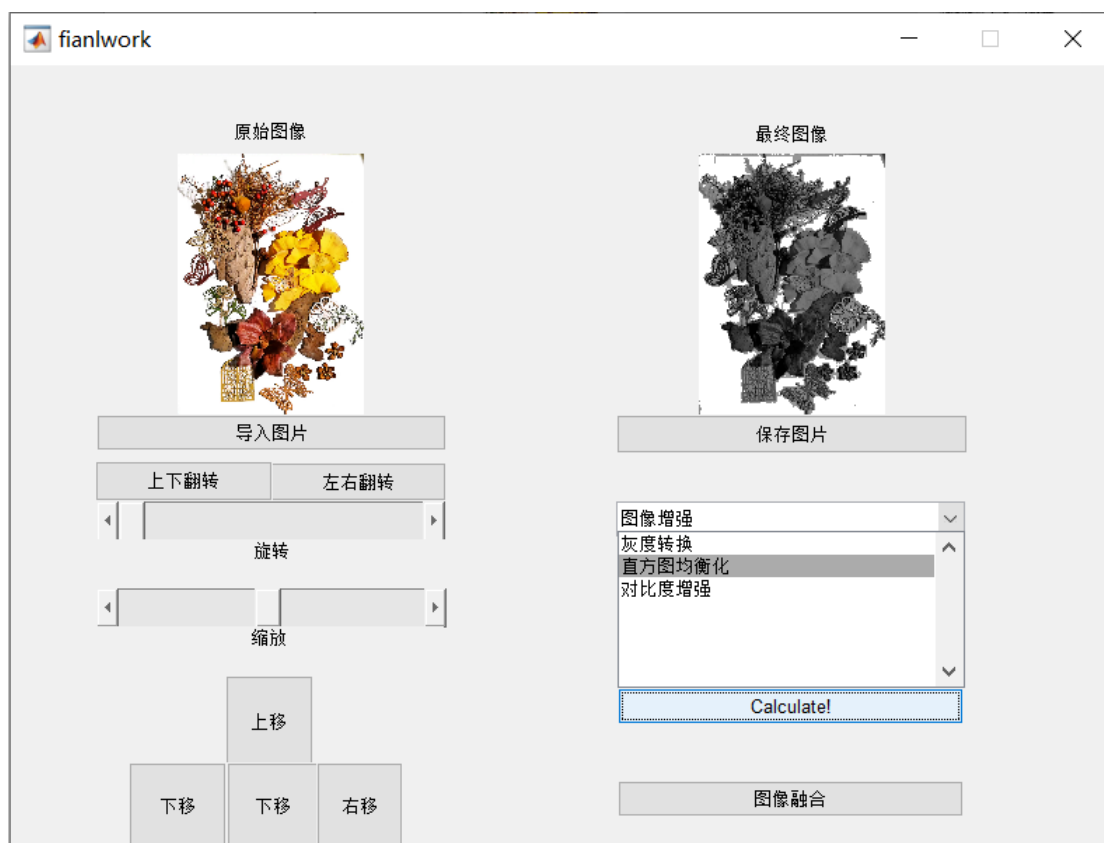
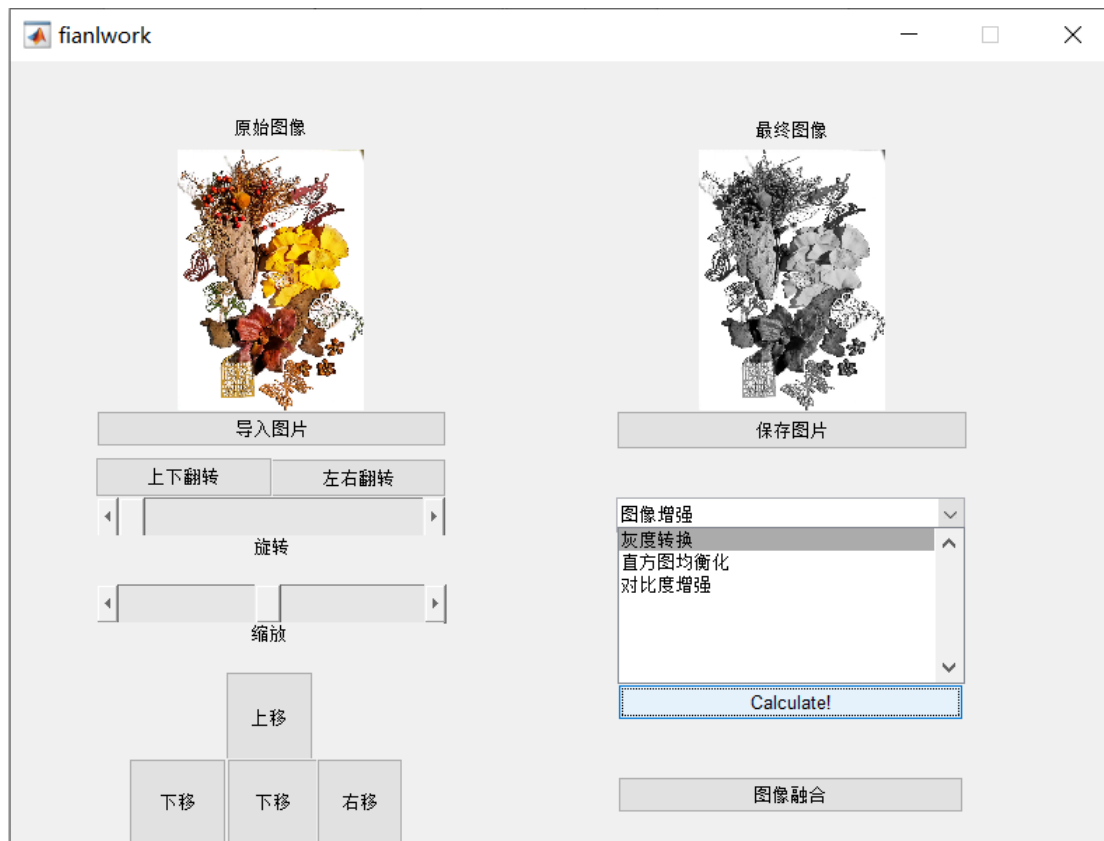


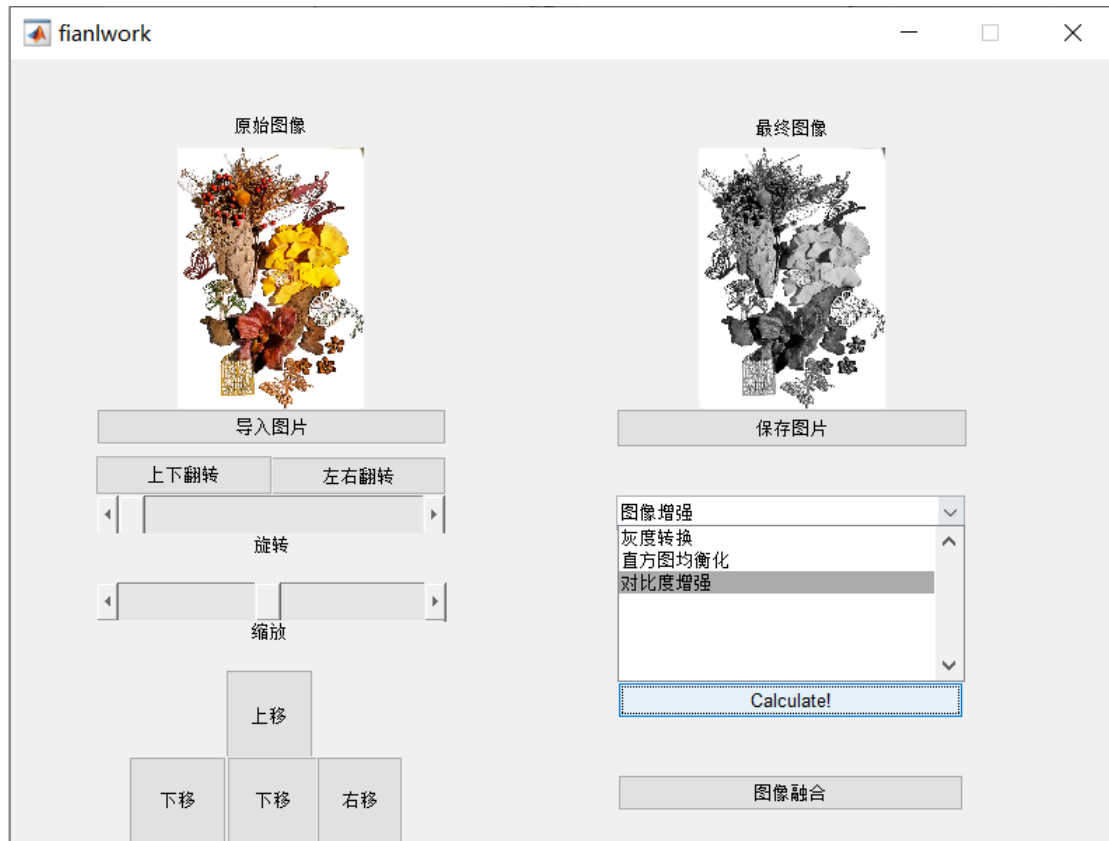
4. 图像滤波



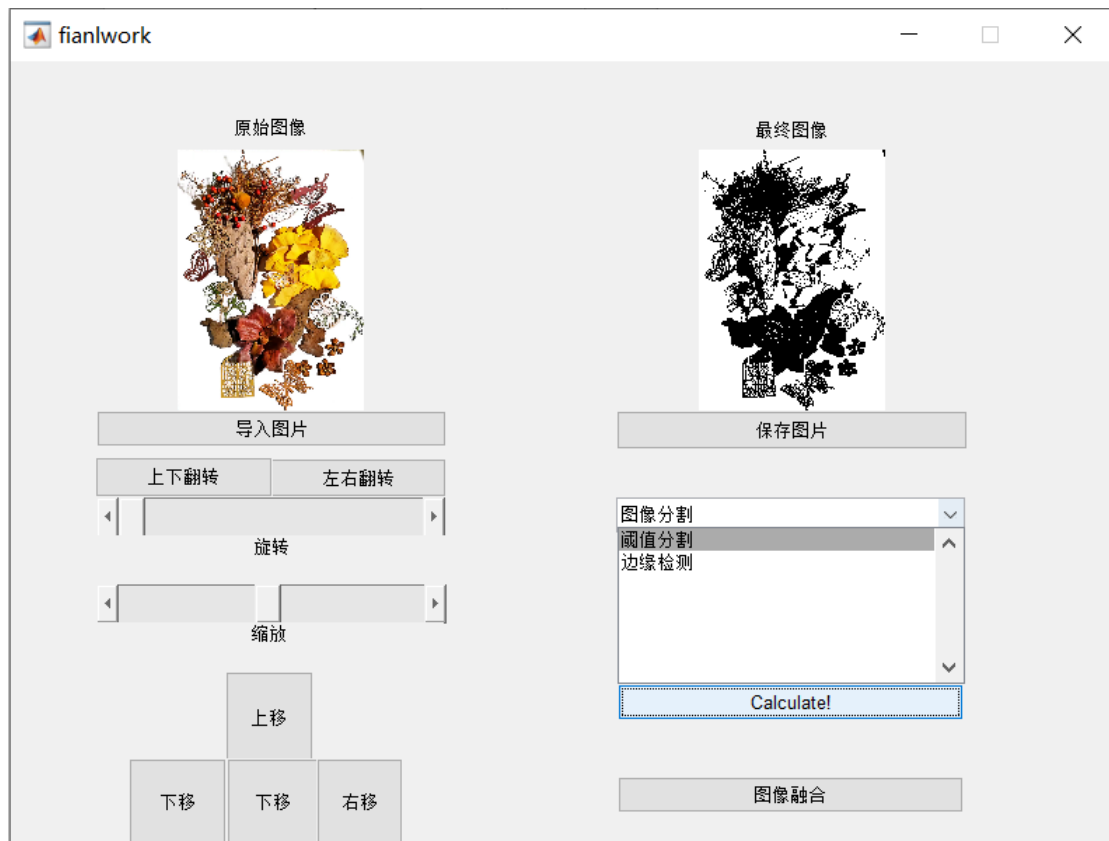


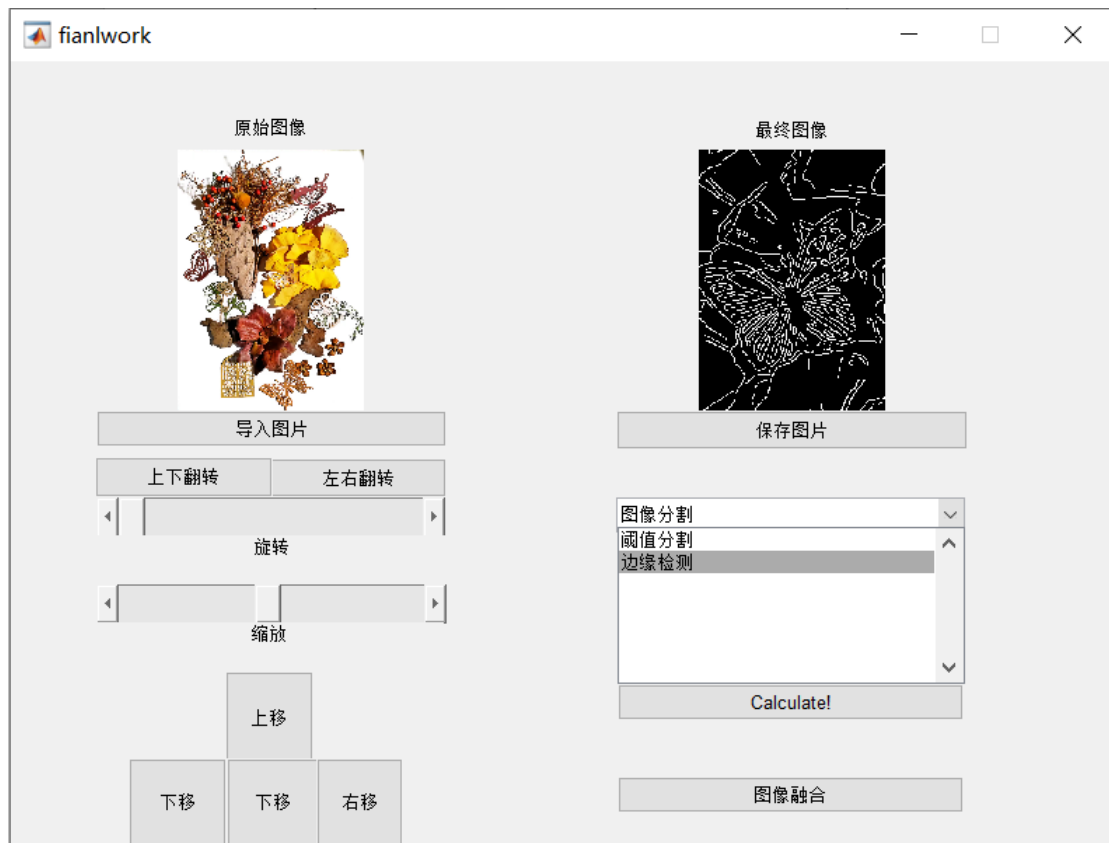
5. 图像增强

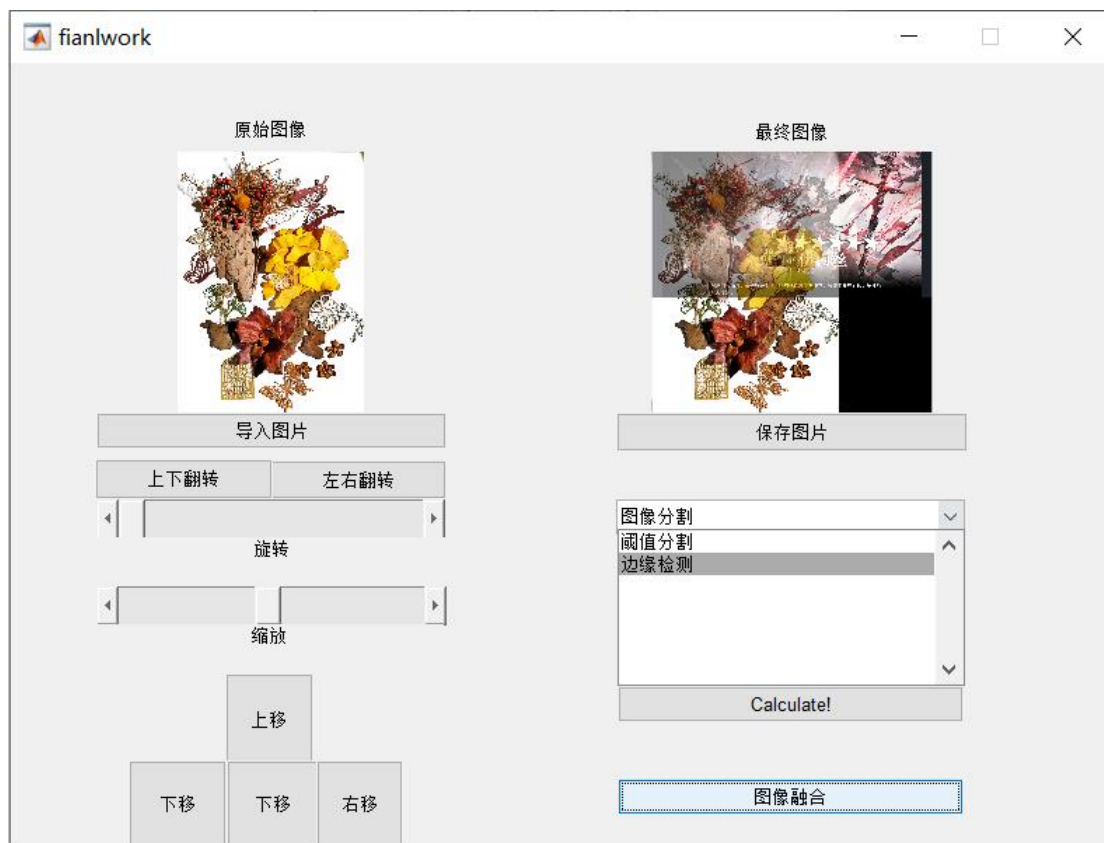




6. 分割功能







三、总结

本程序的创新点是 GUI 设计方面的集成度把控比较好，即不显得拥挤，又比较紧凑，将多种图像处理的功能整合到一起，方便使用者。

优点：

能够提供多种图像处理功能，处理速度快，界面简洁，一目了然。

不足：

部分处理功能的自定义程度还略有不足，只能一键实现功能，应对简单的图像处理场景；平移功能在移动画面到边框外后会将其裁剪，通过我设计的方式移回会无法显示之前移出的部分；部分高级图像处理是调用的是 **matlab** 自带的函数，可能在处理效果上可能更好

遇到的困难

主要是实现缩放时开始坐标轴也会跟着缩放，导致图像分辨率下降但是大小相对没变，后来通过将缩放前的坐标轴保存缩放后将其还原解决。另外就是高级的图像处理的函数不是很清楚，通过在互联网上学习，复现解决（如图像滤波、边缘检测等）但是由于时间有限，没有详细对各种实现方案进行试验，找到最合适的方案，可以后续进行实验，对效果进行提升。