《数字图像处理》实验报告

实验名称　　：实验 4 数字图像的基本运算

实验日期　　：2022.9.23

姓　　名　　：傅康

学　　号　　：084520126

班　　级　　：医信 20

成　　绩　　：

信息技术学院

南京中医药大学

实验目的：

1. 掌握图像的算术运算在数字图像处理中的初步应用。
2. 体会图像算术运算处理的过程和处理前后图像的变化。
3. 体会图像逻辑运算处理的过程和处理前后图像的变化。

实验内容和要求

建立一个名为“xxxxx 实验 4”的解决方案（xxxxx 为自己的学号）

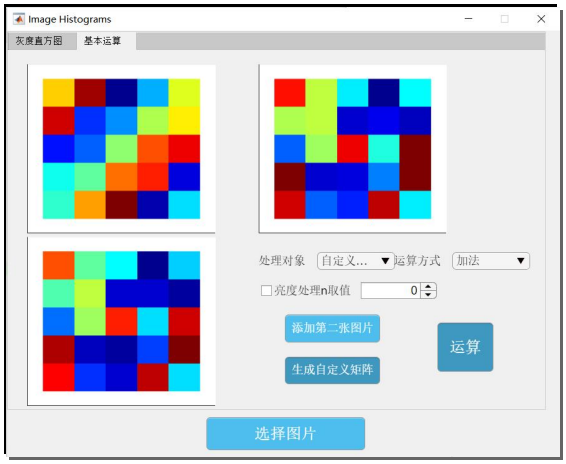
数据准备：两个同样分辨率大小的图像文件，两个 5*5 的二维矩阵

1. 利用 imadd(x,y)函数实现两幅图像及自定义矩阵的加法运算；
2. 利用 imadd(x,n)函数实现图像亮度及自定义矩阵的加法运算；
3. 利用 imsubtract(X,Y)函数实现两幅图像及自定义矩阵的减法运算；
4. 利用 imsubtract （x,n ）函数实现图像及自定义矩阵亮度的减法运算；
5. 利用 immultiply(X,Y)函数实现两幅图像及自定义矩阵的乘法运算；
6. 利用 immultiply （x,n ）函数实现图像及自定义矩阵亮度的乘法运算；
7. 利用 imdivide(X,Y)函数实现两幅图像及自定义矩阵的除法运算；
8. 利用 imdivide(x,n ） 函数实现图像及自定义矩阵亮度的除法运算；
9. 实现图像及自定义矩阵的三种逻辑运算。
10. 分别利用 imhis()函数、bar()函数、stem()函数实现图像灰度直方图的显示；

运行结果（写清题号）

描述实验的基本步骤，用数据和图片给出各个步骤中取得的实验结果和源代码，并进行必要的讨论，必须包括原始图像及其计算处理后的图像以及相应的解释。

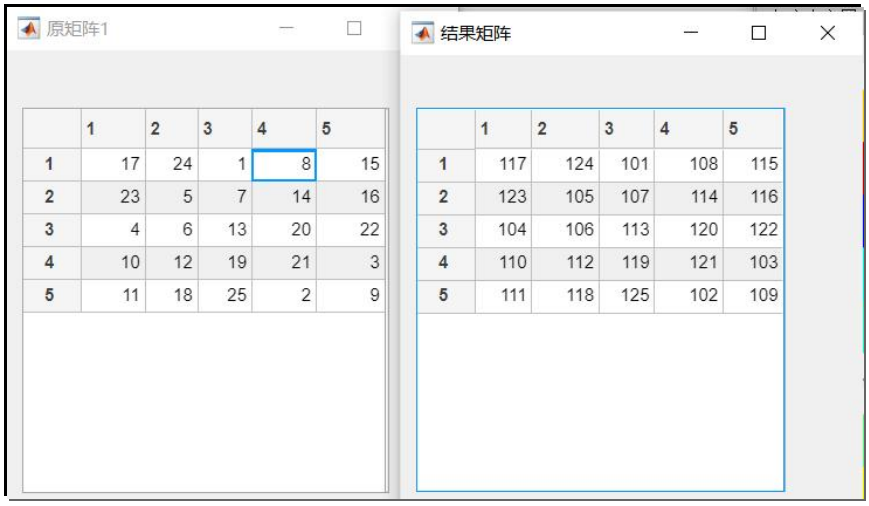
1. 利用 imadd(x,y)函数实现两幅图像及自定义矩阵的加法运算；



矩阵运算结果

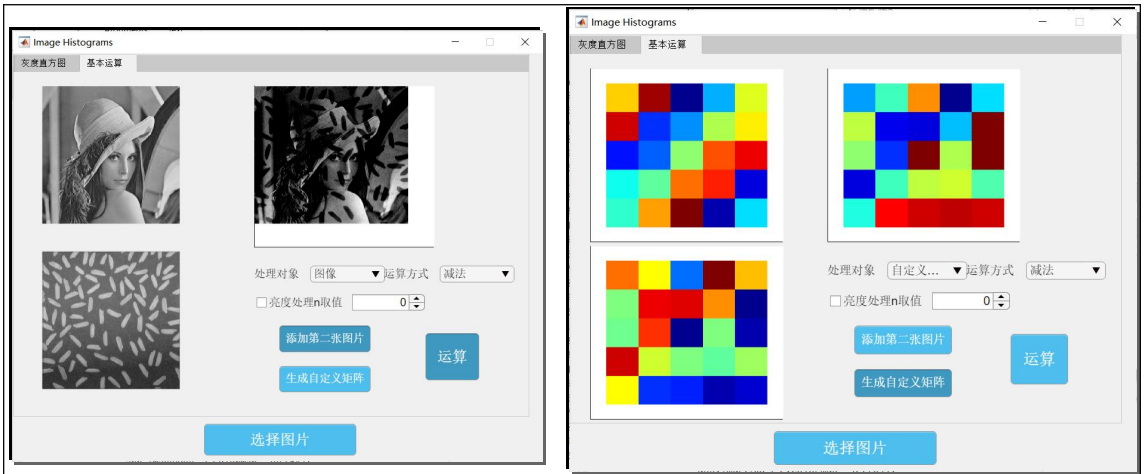
以[1，1]为例，17+207=224，图像加法是矩阵对应元素相加。

2. 利用 imadd(x,n)函数实现图像**亮度**及自定义矩阵的加法运算；



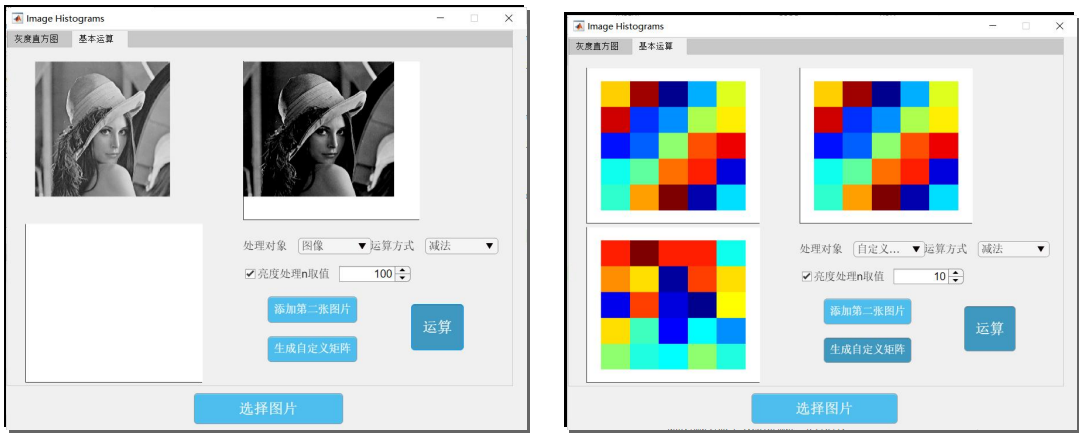
以[1，1]为例，17+100=117，亮度的加法运算和图像同理。

3. 利用 imsubtract(X,Y)函数实现两幅图像及自定义矩阵的减法运算；

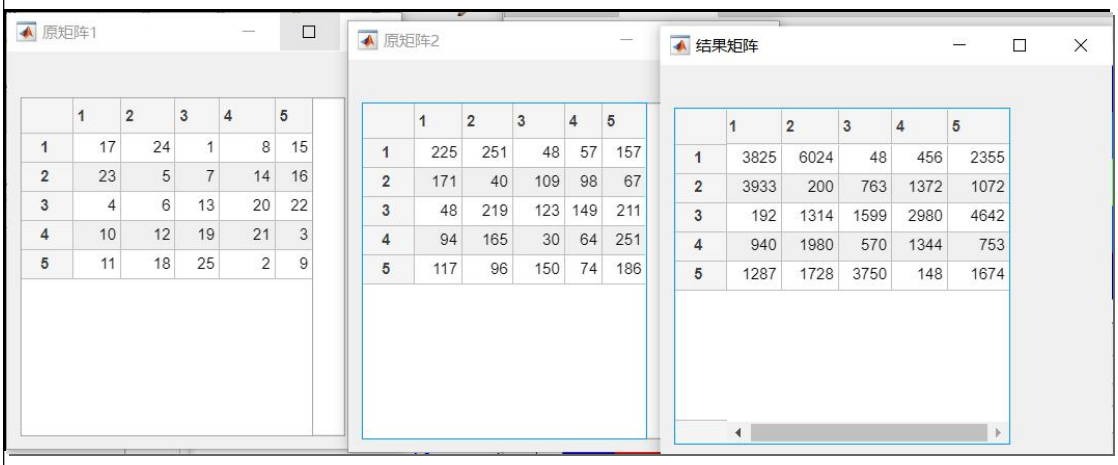


以[1，1]为例，17-190=-173，图像减法是矩阵对应元素相减，下面亮度减法同理。

4. 利用 imsubtract（x, n ）函数实现图像及自定义矩阵亮度的减法运算；

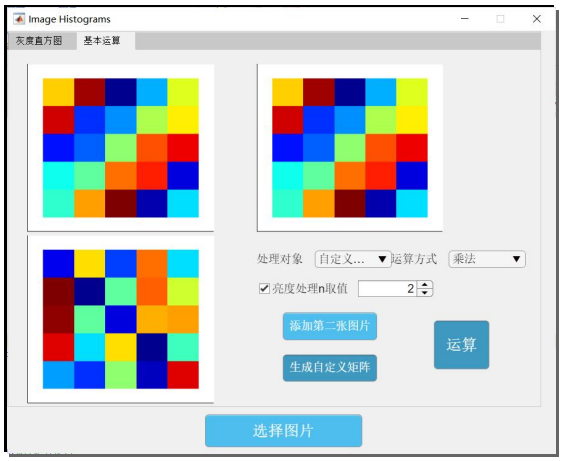


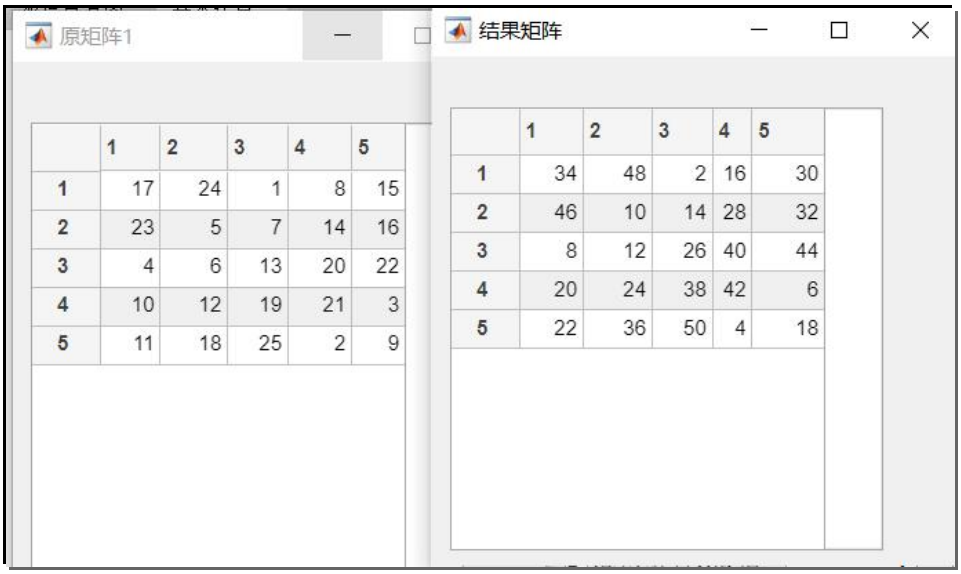
5. 利用 immultiply(X,Y)函数实现两幅图像及自定义矩阵的乘法运算；



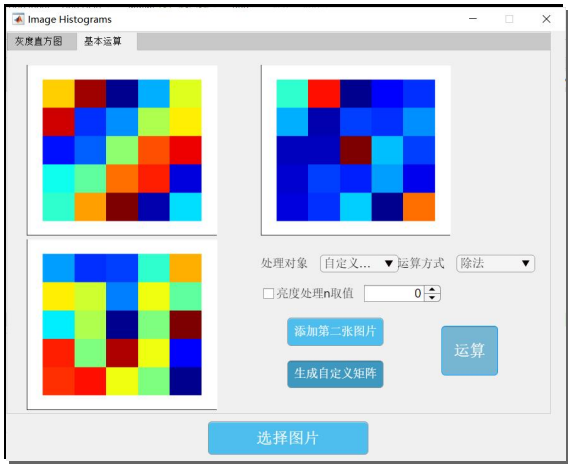
以[1，1]为例，17×225=3825，图像乘法是矩阵对应元素相乘，下面亮度乘法同理。

6. 利用 immultiply（x, n ）函数实现图像及自定义矩阵亮度的乘法运算；



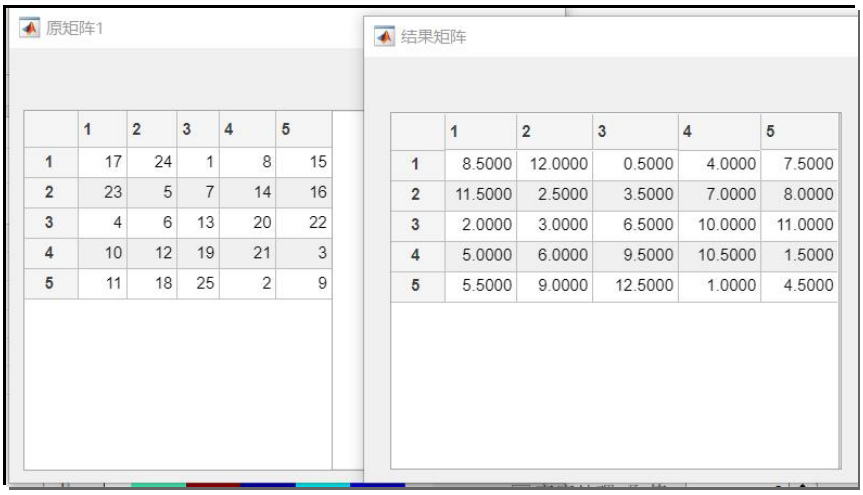


7. 利用 imdivide(X,Y)函数实现两幅图像及自定义矩阵的除法运算；



以[1，1]为例，17÷89≈0.1910，图像除法是矩阵对应元素相除，下面亮度除法同理。

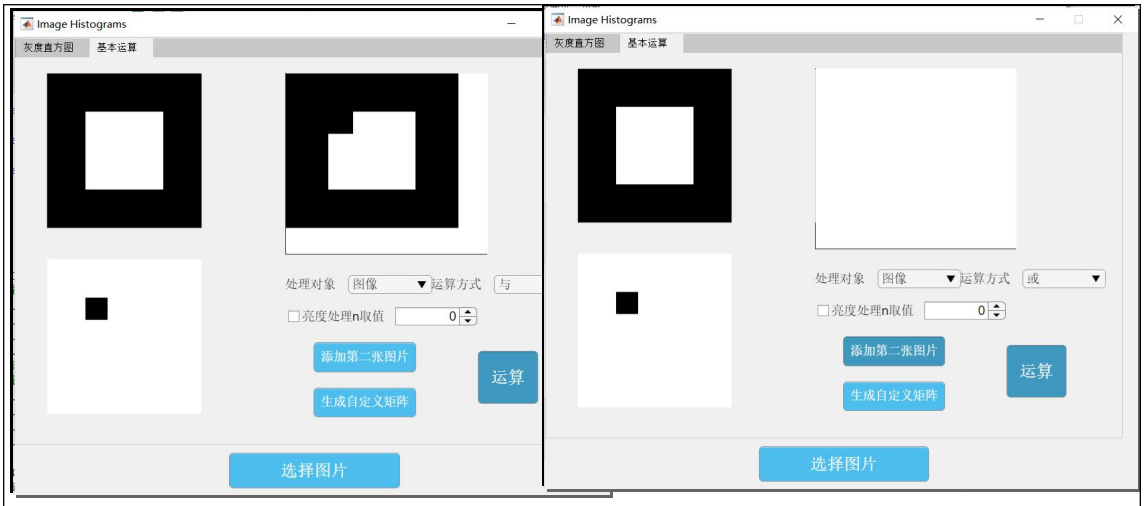
8. 利用 imdivide(x, n) 函数实现图像及自定义矩阵亮度的除法运算；

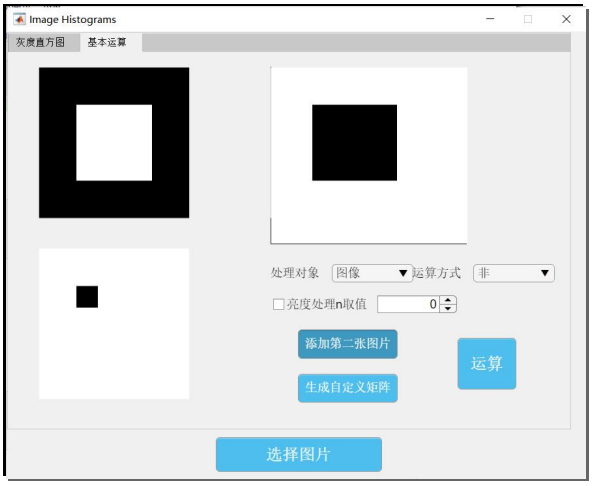


基本运算代码：

```
function ButtonValueChanged(app, event)
            switch app.DropDown_3.Value
                case '加法'
                    if app.nCheckBox.Value==1%亮度处理
                        app.imC=imadd(app.imA,app.nSpinner.Value,'uint16');
                    else%两图像/矩阵处理
                        app.imC=imadd(app.imA,app.imB,'uint16');
                    end
                case '减法'
                    if app.nCheckBox.Value==1
                        app.imC=imsubtract(app.imA,app.nSpinner.Value);
                    else
                        app.imC=imsubtract(app.imA,app.imB);
                    end
                case '乘法'
                    if app.nCheckBox.Value==1
                        app.imC=immultiply(app.imA,app.nSpinner.Value);uint16(app.imC);
                    else
                        app.imC=immultiply(app.imA,app.imB);uint16(app.imC);
                    end
                otherwise%除法
                    if app.nCheckBox.Value==1
                        app.imC=imdivide(app.imA,app.nSpinner.Value);
                    else
                        app.imC=imdivide(app.imA,app.imB);
                    end
            end
            if app.DropDown_2.Value=="图像"%图像
                %设置灰度的颜色表
                app.ImageAxes.Colormap = gray(256);
                app.ImageAxes_2.Colormap=gray(256);
                app.ImageAxes_3.Colormap=gray(256);
                app.ImageAxes_4.Colormap=gray(256);
            else%自定义矩阵
                %设置彩色的颜色表
                app.ImageAxes.Colormap =  jet(64);
                app.ImageAxes_2.Colormap= jet(64);
                app.ImageAxes_3.Colormap= jet(64);
                app.ImageAxes_4.Colormap= jet(64);
            end
            imagesc(app.ImageAxes_4,app.imC);
            fig=uifigure('Name','结果矩阵');
            uitable('Parent',fig,"Data",app.imC);
        end
```

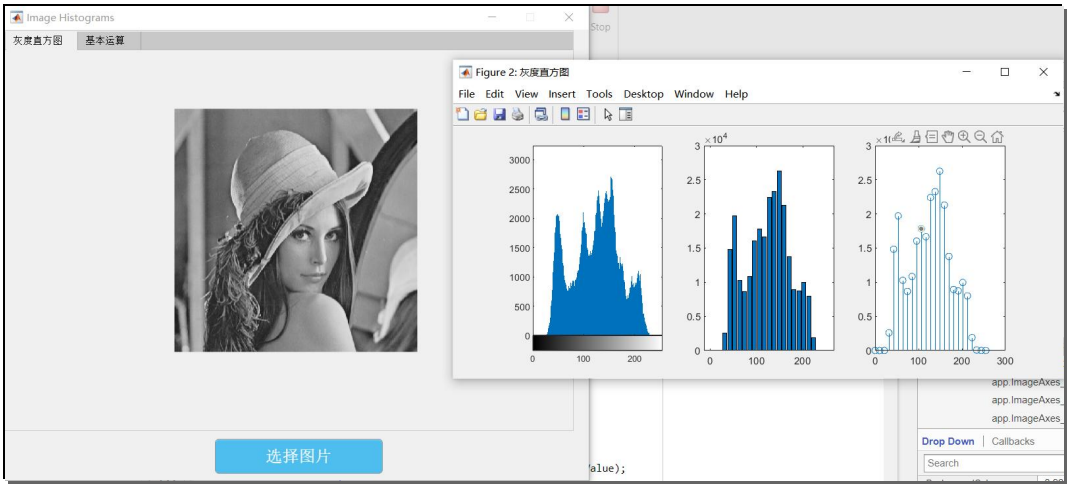
9. 实现图像及自定义矩阵的三种逻辑运算（**二值图像**）。






```
case '与'
    app.imC=app.imA&app.imB;
case '或'
    app.imC=app.imA|app.imB;
case '非'
    app.imC=~app.imA;
```

10. 分别利用 imhist()函数、bar()函数、stem()函数实现图像灰度直方图的显示；



```
function LoadButton_2Pushed(app, event)
    % Display uigetfile dialog
    filterspec = {'*.jpg;*.tif;*.png;*.gif;*.bmp','All Image Files'};
    [f, p] = uigetfile(filterspec);
    if isequal(f,0) || isequal(p,0)
        errordlg("没有选中图片文件",'错误');
    else
       app.file=strcat(p,f);
       app.imA=imread(app.file);
       imshow(app.imA,'Parent',app.ImageAxes,'InitialMagnification',100);
       imshow(app.imA,'Parent',app.ImageAxes_2,'InitialMagnification',100);
       figure("Name",'灰度直方图');
       subplot(1,3,1);
       imhist(app.imA);
       h=imhist(app.imA,25);%把255个灰度值分成25段
       horz=linspace(0,255,25);
       subplot(1,3,2);
       bar(horz,h);
       subplot(1,3,3)
       stem(horz,h);
    end
end
```

讨论/注意

①两灰度图相加超过 255 可用 uint16 防止图像信息丢失，相减小于 0 的处理方式有两种：截断为 0，但新图像像素值偏小，整体会变暗；而是取绝对值。

②为何二值模板无法对原图像进行局部显示和提取？

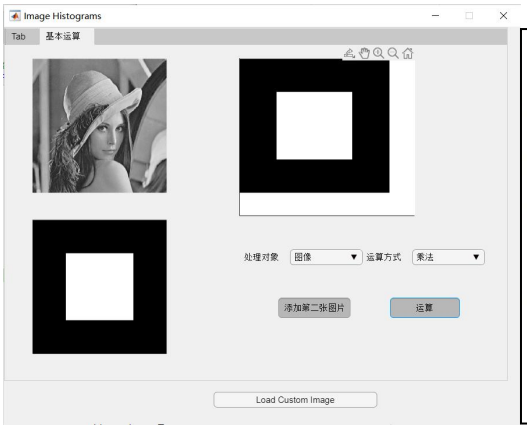
因为 0 乘任何数都为 0，1 乘任何数都等于其本身，利用这个性质，图像的乘法运算可以用二值模板图像对特定的区域进行图像的局部显示与提取。但我在尝试的时候发现并未对图像进行局部提取，后来经检测发现我生成的是“二值”的灰度图像，即白色是 255 而不是 1 所以运算出来的是超过 255 的白色而不是本身，所以后面我把模板矩阵二值化后解决此问题。注：

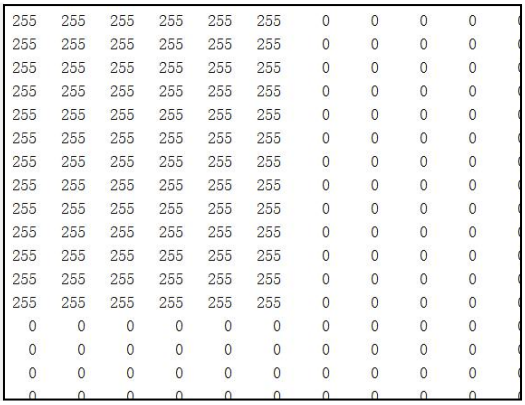
% 作为变量，二值图像不需要后缀，类名为‘logical’；

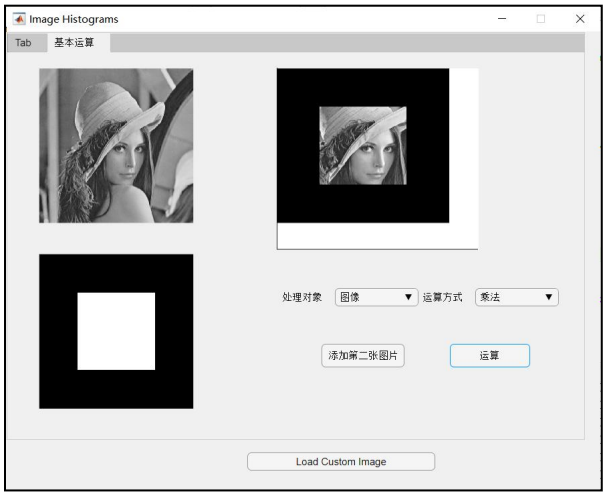
% 作为图像文件，二值图像可以保存为任何图像格式，没有专门的后缀指定。

```
A=imbinarize(A);
```

二值化







③指定 imshow 的父坐标系 imshow(app.imC,'Parent',app.ImageAxes_4);

④imagesc 显示矩阵，若由灰度要求，可以设置其颜色表 colormap 为 gray(256)；imshow 显示图像。

实验的体会与思考题

1. 由图像算术运算的运算结果，思考图像减法运算在什么场合上发挥优势？

①显示两幅图像的差异，可以检测同一场景两幅图像之间的变化。

②去除不需要的叠加性图案。比如周期性的噪声等。

③图像分割。比如一个房间的静止照片 1，和有一个人的房间静止照片 2，两照片相减就能把人给分割出来，把房间背景去除。

④生成合成图像。