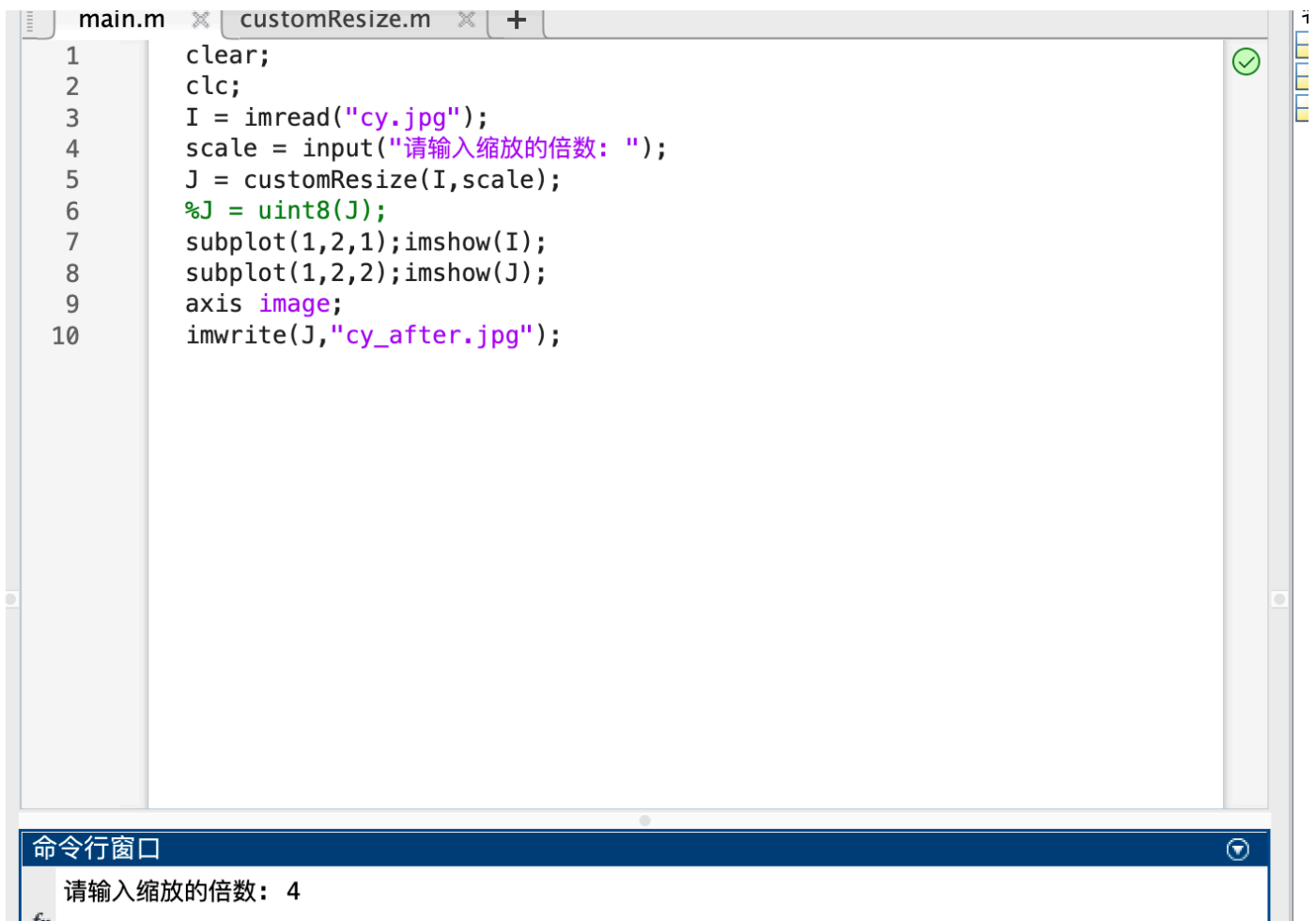


一、启动main.m函数



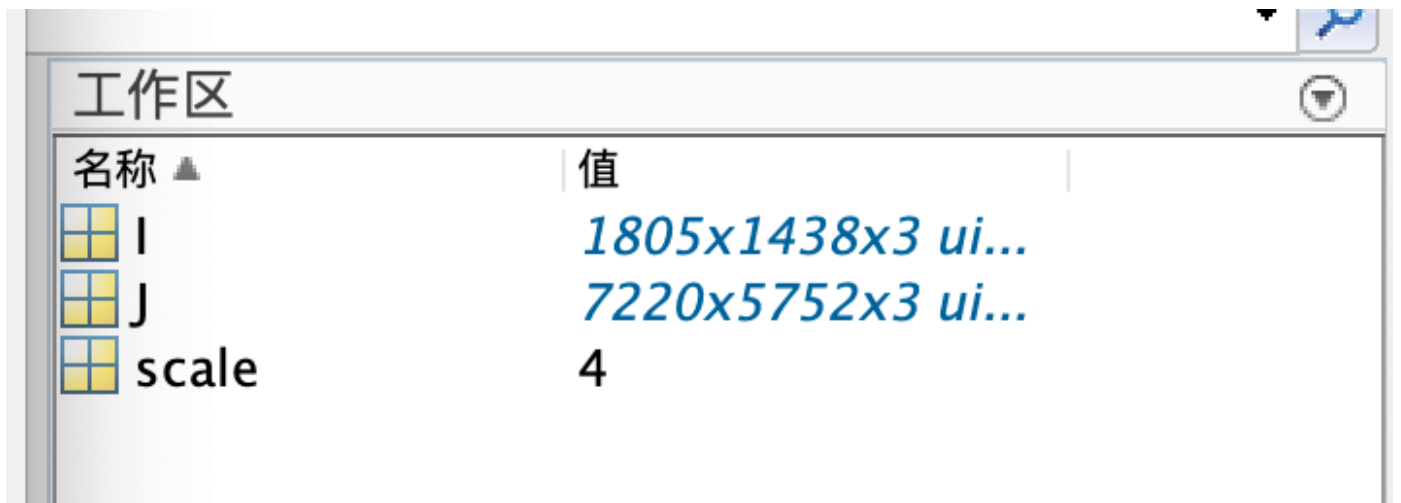
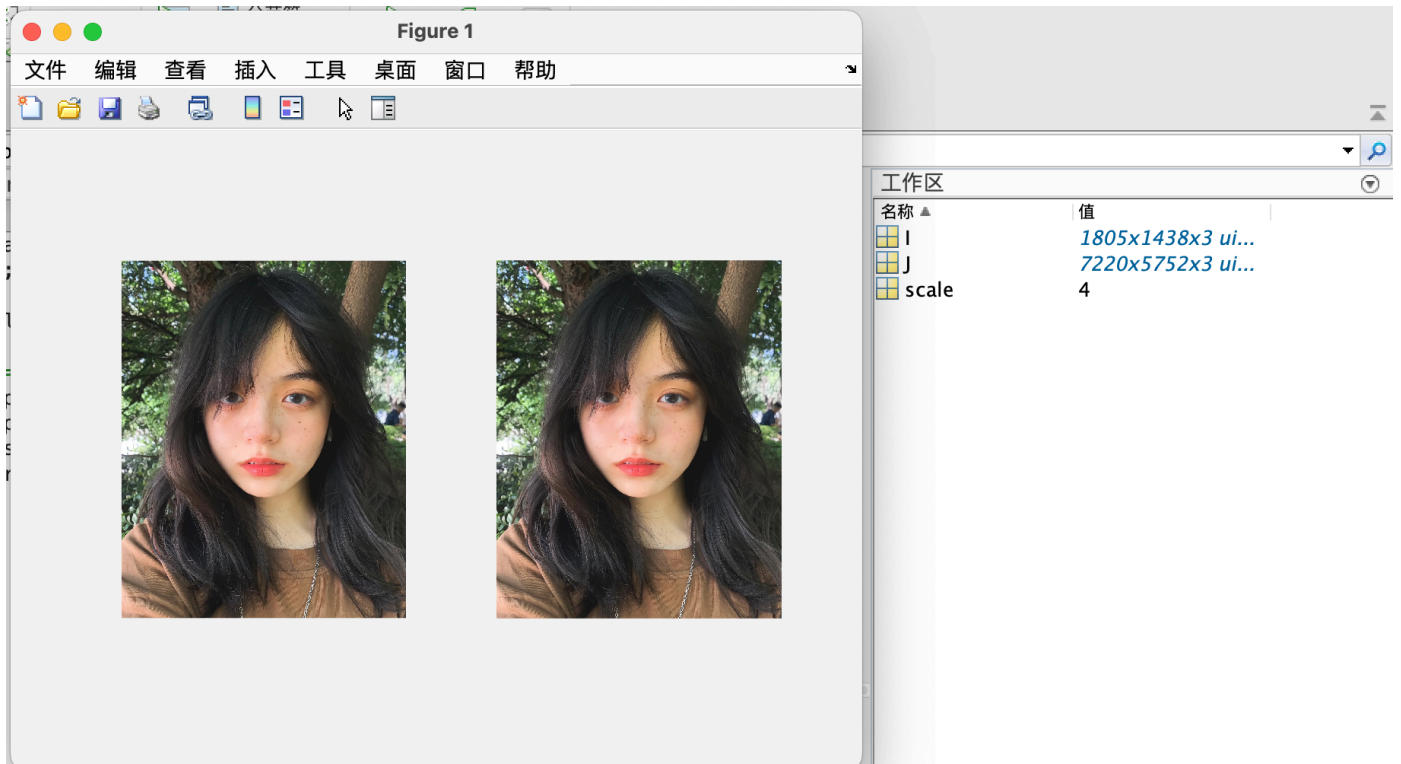
The image shows a MATLAB editor window with two tabs: 'main.m' and 'customResize.m'. The 'main.m' tab is active, displaying the following code:

```
1 clear;
2 clc;
3 I = imread("cy.jpg");
4 scale = input("请输入缩放的倍数: ");
5 J = customResize(I, scale);
6 %J = uint8(J);
7 subplot(1,2,1);imshow(I);
8 subplot(1,2,2);imshow(J);
9 axis image;
10 imwrite(J, "cy_after.jpg");
```

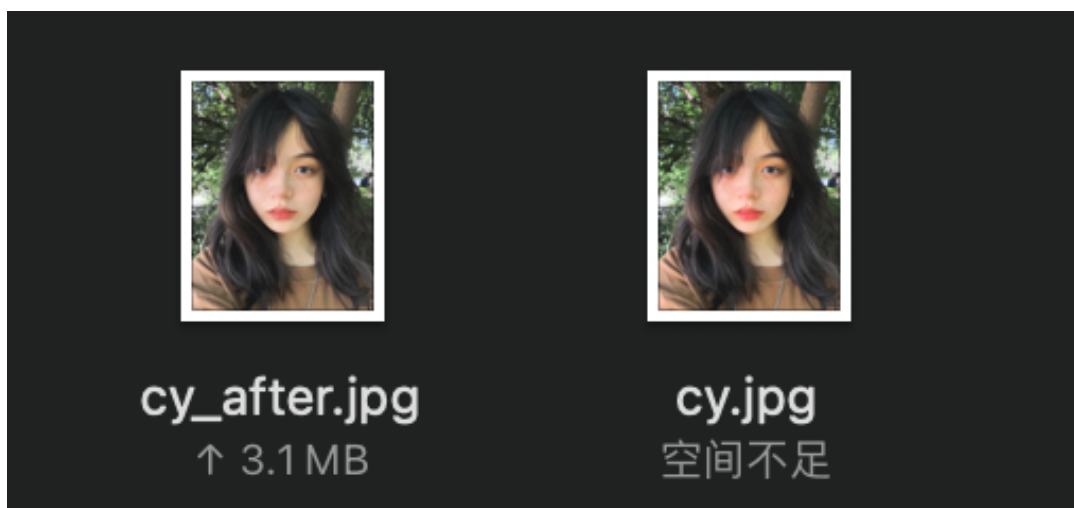
Below the editor is the '命令行窗口' (Command Window). It displays the prompt '请输入缩放的倍数: 4', indicating that the user has entered the value 4.

输入缩放的倍数

二、实验结果

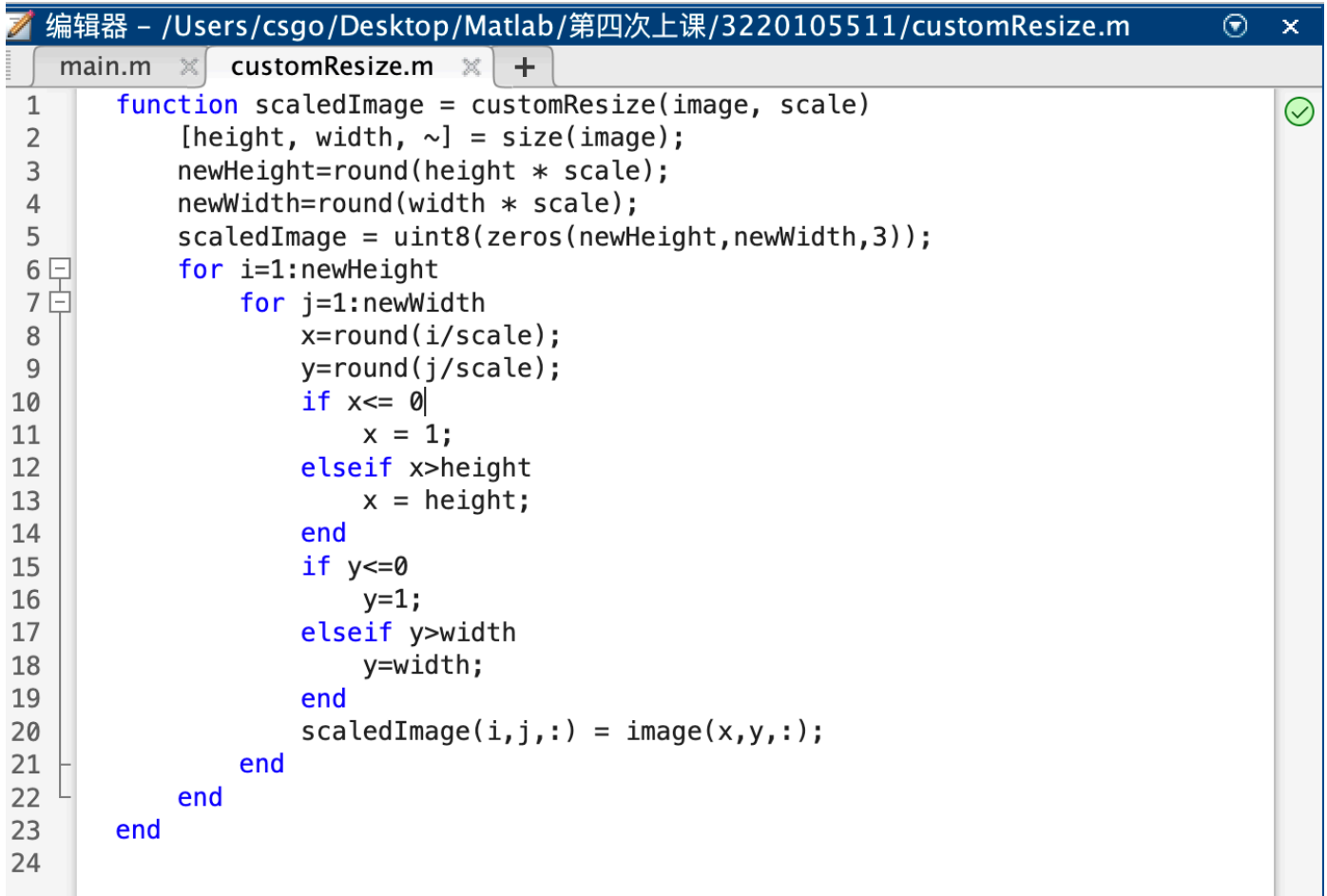


观察到已经实现了缩放功能



并且在imwrite之后生成了两张图片

三、本函数所使用的图片放缩功能

A screenshot of a MATLAB editor window. The title bar reads '编辑器 - /Users/csgo/Desktop/Matlab/第四次上课/3220105511/customResize.m'. There are two tabs: 'main.m' and 'customResize.m'. The 'customResize.m' tab is active, showing the following code:

```
1 function scaledImage = customResize(image, scale)
2     [height, width, ~] = size(image);
3     newHeight=round(height * scale);
4     newWidth=round(width * scale);
5     scaledImage = uint8(zeros(newHeight,newWidth,3));
6     for i=1:newHeight
7         for j=1:newWidth
8             x=round(i/scale);
9             y=round(j/scale);
10            if x<= 0
11                x = 1;
12            elseif x>height
13                x = height;
14            end
15            if y<=0
16                y=1;
17            elseif y>width
18                y=width;
19            end
20            scaledImage(i,j,:) = image(x,y,:);
21        end
22    end
23 end
24
```

The code is a function that takes an image and a scale factor as input and returns a scaled image. It uses nested loops to iterate over the new dimensions and calculates the corresponding pixel coordinates in the original image. The code is syntactically correct, as indicated by a green checkmark in the top right corner of the editor window.

对于每个目标图像的像素位置 (i, j) ，通过计算原图像中的相应位置 (x, y) ，然后根据邻近的四个像素的权重进行插值计算。