数字图像处理　Problem5(2)

1552746 崔鹤洁　　Date: 2017-10-10

(2) Give one intensity transformation function forspreading the intensities of an image such that the lowestis Imin and the highest is Imax , (0Imin Imax 255). Denote by Fmax and Fmin the maximum and minimum intensities values of the input image.

✥**result image:**

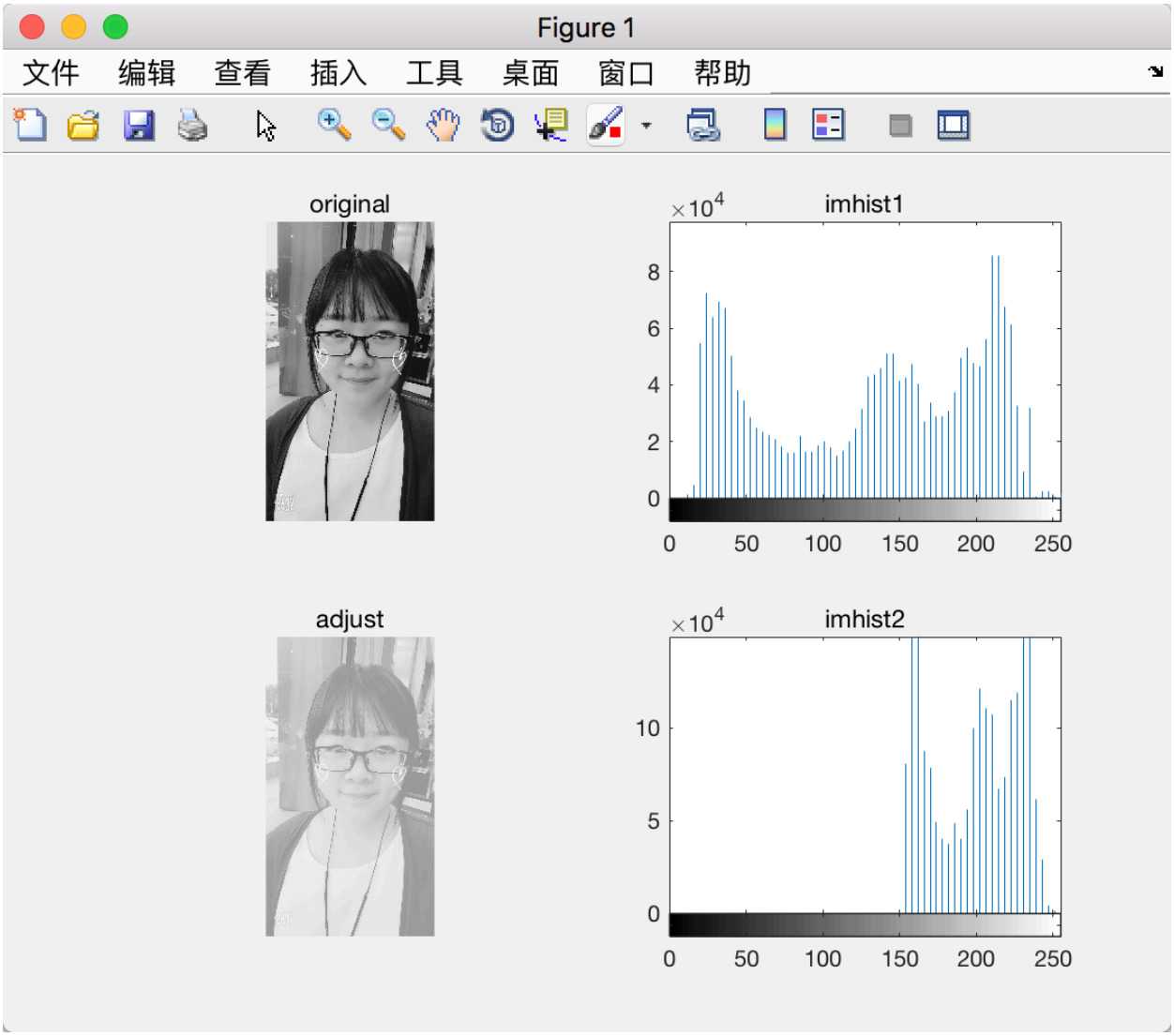
Analysis:

Running the program, it will first let the user input the target range of intensities values(0~255);

Then, it will cast the maximum and minimum intensities values of the input image to the target.

In the code part, I provide 2 implements, one is using Matlab function ***imadjust***, the other is a linear mapping (sigma=1) without using Matlab Function.

When you set the range from **150 to 250**, the result is shown as below:



✥code:

Here I provide two implement code using different methods:

- Impletment Matlab function ***imadjust:***

```
clear;
img=imread('cui.jpg');
img_gray = rgb2gray(img);

% let the user input the target range of intensities values
targetMax = input('Please input target I_max(0~255): ');
targetMin = input('Please input target I_min(0~255): ');
% Change the target range from 0 to 1 to use the adjust function
targetMax = targetMax/256;
targetMin = targetMin/256;

% find the maximum and minimum intensities values of the input image
counts = imhist(img_gray);
minBinValue = find(counts>0, 1, 'first');
minBinValue = minBinValue/256;
maxBinValue = find(counts>0, 1, 'last');
maxBinValue = maxBinValue/256;

% spreading the intensities of the input image
img_adj = imadjust(img_gray,[minBinValue,maxBinValue],
[targetMin,targetMax]);

figure;

subplot(2,2,1);
imshow(img_gray);
title('original')

subplot(2,2,2);
imhist(img_gray,64);
title('imhist1')

subplot(2,2,3);
imshow(img_adj);
title('adjust')

subplot(2,2,4);
imhist(img_adj,64);
title('imhist2')
```

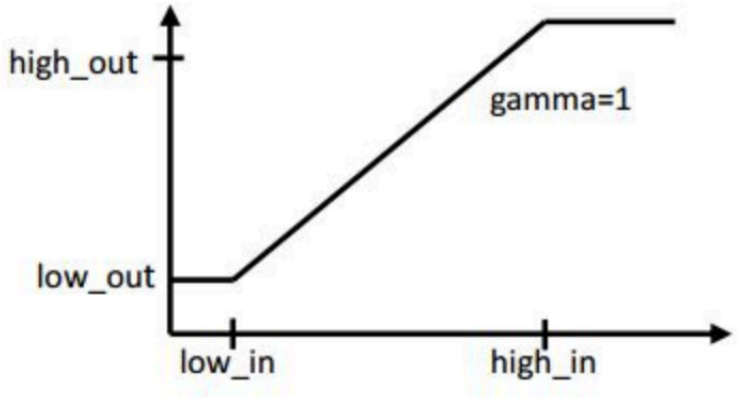
- Impletment ***without*** Matlab function ***imadjust***:

Analysis:

Suppose *** J = imadjust(I,[low_in; high_in],[low_out; high_out],gamma)***

At the first, I read from a algorithm from CSDN, which is described as below: Map the brightness value in image I to the new value in J, and the value between low_in and hige_in is mapped to the value between low_out and high_out. The values below low_in and above are cut off, which means that the values below low_in are mapped to low_out, and the values above high_in are mapped to high_out.

The gray mapping process is shown as the followed picture:



However, when I finished in the way above and calculate the imhist of this image, it shows a different imhist from the way using imadjust, so I changed the algorithm in this way:

$$img\_adjust(i,j) = targetMin + img\_gray(i,j) * \frac{targetMax - targetMin}{256}$$

Then I show the imhist of this processed picture, it is the same as the imhist of using **imadjust**

```
clear;
img=imread('cui.jpg');
img_gray = rgb2gray(img);
img_adj = img_gray;

% let the user input the target range of intensities values
targetMax = input('Please input target I_max(0~255): ');
targetMin = input('Please input target I_min(0~255): ');

new_range = targetMax-targetMin;

% Linear projection imadjust
[m,n] = size(img_gray);
for i = 1:m
    for j = 1:n
        img_adj(i,j) = targetMin+img_gray(i,j)*(new_range/256);
    end
end

figure;

subplot(2,2,1);
imshow(img_gray);
title('original')

subplot(2,2,2);
imhist(img_gray,64);
title('imhist1')

subplot(2,2,3);
imshow(img_adj);
title('adjust')

subplot(2,2,4);
imhist(img_adj,64);
title('imhist2')
```