

作业 1: 开发你自己的"美图"软件

姓名 王明 学号 161220124 邮箱 1090616897@qq.com 联系方式 13851085654

(南京大学 计算机科学与技术系, 南京 210093)

1 实现细节

1.1 灰度图

1.1.1 思路

灰度图只有一个通道, 即 0-255 的灰度通道。那么可以直接使用直方图均衡化的方法, 将原本像素点不均衡的灰度级分布转换成相对来说较为均衡的分布。核心思想是得到灰度转换映射函数。使用如下图所示的累积分布函数, 即可完成灰度的转换。(其中 L 是灰度级的数目, A_0 是像素点的总数)

$$f(D_A) = \frac{L}{A_0} \sum_{u=0}^{D_A} H_A(u)$$

1.1.2 实现

首先遍历输入的图像, 计数得到直方图, 然后采用累积分布函数进行灰度的转换得到确定的灰度转换映射函数, 最后再遍历输入的图像, 根据转换函数得到转换后的图像。

```
function [output2] = hist_equal(input_channel)
%concretely realize hist_qual
%get basic information of input
[r,c] = size(input_channel);
total_pixel = r * c;
total_graylevel = 256;
%count dots for each graylevel
hist_a = zeros(1,total_graylevel);
for i = 1:r
    for j = 1:c
        hist_a(1,input_channel(i,j)+1) = hist_a(1,input_channel(i,j)+1) + 1;
    end
end
hist_a = hist_a / total_pixel;
%get the graylevel conversion function
out_b = zeros(1,total_graylevel);
for len = 1:total_graylevel
    sum = 0;
    for tmp = 1:len
        sum = sum + hist_a(1,tmp);
    end
    out_b(1,len) = round(sum * (total_graylevel - 1));
end
%apply graylevel conversion function to tranform original input
for i = 1:r
    for j = 1:c
        input_channel(i,j) = out_b(1,input_channel(i,j)+1);
    end
end
output2 = input_channel;
end
```

1.2 RGB彩色图

1.2.1 思路 1

模仿灰度图的做法，使用 `hist_equal()` 分别对 RGB 三个通道进行处理，然后再整合处理后的 RGB 得到最终的图像。

1.2.2 实现 1

```
function [output] = method1(input_image)
%apply hist_equal to R/G/B separately
r=input_image(:,:,1);
v=input_image(:,:,2);
b=input_image(:,:,3);
r1 = hist_equal(r);
v1 = hist_equal(v);
b1 = hist_equal(b);
output = cat(3,r1,v1,b1);
end
```

1.2.3 思路 2

从思路 1 的实现效果和逻辑上来看，思路 1 是存在问题的。实现效果上，思路 1 处理后的图片失真的情况很严重，不是增强了图像，而是很大程度上改变了图像。逻辑上，将 RGB 三个通道分开处理是不科学的，因为三种颜色各自的分布不一样，对于同一个像素点，映射之后 RGB 对应的变化程度是不一致的，因此很容易失真。

由此必须对于总体的值进行均衡化，由此想到将 RGB 转成其他形式，再进行均衡化处理。

这里使用了 3 种模型如下：

- 1、先转换成 HSV，然后对于 V（亮度）进行均衡化而增强图像。
- 2、先转换成 HSI，然后对于 I（亮度）进行均衡化而增强图像。
- 3、先转换成 YUV，然后对于 Y（亮度）进行均衡化而增强图像。
- 4、先转换成 YIQ，然后对于 Y（亮度）进行均衡化而增强图像。
- 5、先转换成 YCbCr，然后对于 Y（亮度）进行均衡化而增强图像。

1.2.4 实现 2

```
function [output] = method2(input_image)
%convert RGB to HSV, apply hist_equal to V-dimension
tmp = rgb2hsv(input_image);
h = tmp(:,:,1);
s = tmp(:,:,2);
v = tmp(:,:,3);
h1 = h;
s1 = s;
v1 = hist_equal(round(v * 255))/255;
output = hsv2rgb(cat(3,h1,s1,v1));
end
```

```
function [output] = method3(input_image)
%convert RGB to HSI, apply hist_equal to I-dimension
tmp = rgb2hsi(input_image);
h = tmp(:, :, 1);
s = tmp(:, :, 2);
i = tmp(:, :, 3);
h1 = h;
s1 = s;
i1 = hist_equal(round(i * 255))/255;
output = hsi2rgb(cat(3,h1,s1,i1));
end
```

```
function [output] = method4(input_image)
%convert RGB to YUV, apply hist_equal to Y-dimension
tmp = rgb2yuv(input_image);
y = tmp(:, :, 1);
u = tmp(:, :, 2);
v = tmp(:, :, 3);
y1 = hist_equal(round(y * 255))/255;
u1 = u;
v1 = v;
output = yuv2rgb(cat(3,y1,u1,v1));
end
```

```
function [output] = method5(input_image)
%convert RGB to YIQ, apply hist_equal to Y-dimension
tmp = rgb2ntsc(input_image);
y = tmp(:, :, 1);
i = tmp(:, :, 2);
q = tmp(:, :, 3);
y1 = hist_equal(round(y * 255))/255;
i1 = i;
q1 = q;
output = ntsc2rgb(cat(3,y1,i1,q1));
end
```

```
function [output] = method6(input_image)
%convert RGB to YCbCr, apply hist_equal to Y-dimension
tmp = rgb2ycbcr(input_image);
y = tmp(:, :, 1);
cb = tmp(:, :, 2);
cr = tmp(:, :, 3);
y1 = hist_equal(round(y * 255))/255;
cb1 = cb;
cr1 = cr;
output = ycbcr2rgb(cat(3,y1,cb1,cr1));
end
```

2 结果

2.1 实验设置

输入: jpg 图像 (灰度图或者 RGB 彩色图)。

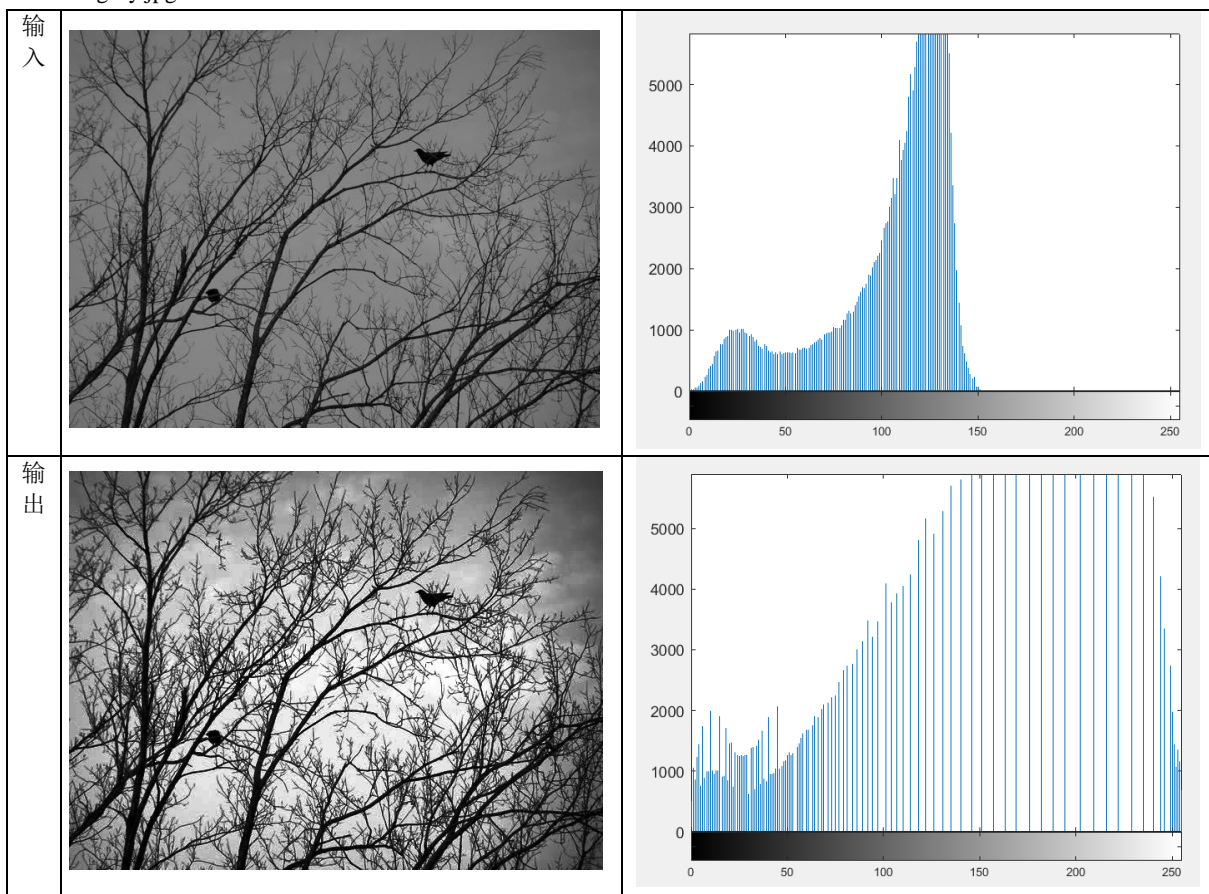
输出: jpg 图像 (直方图均衡化后的图像)。

额外说明: 对于彩色图, 采用 6 种方法处理 (需要手动选择, 默认使用 method3)。同时输出了灰度直方

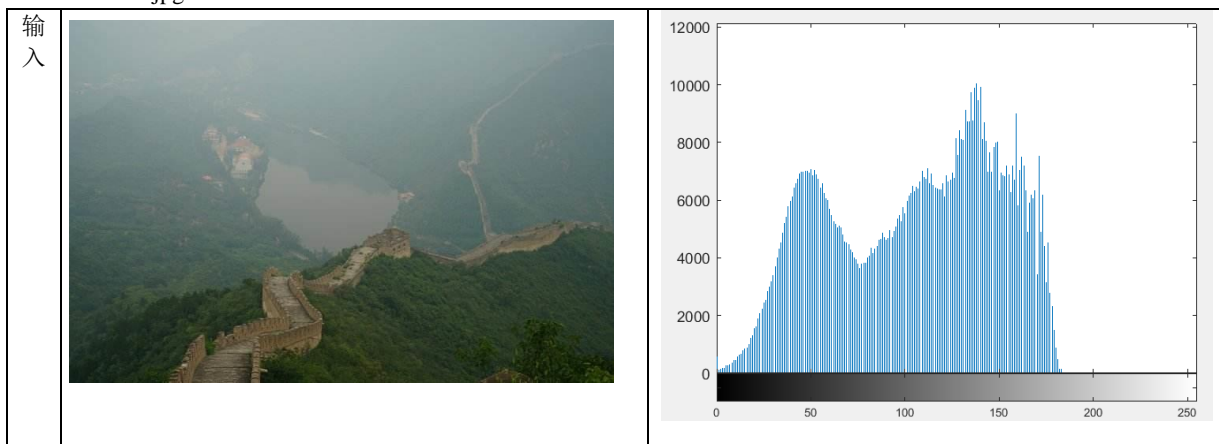
图，方便结果的对比。

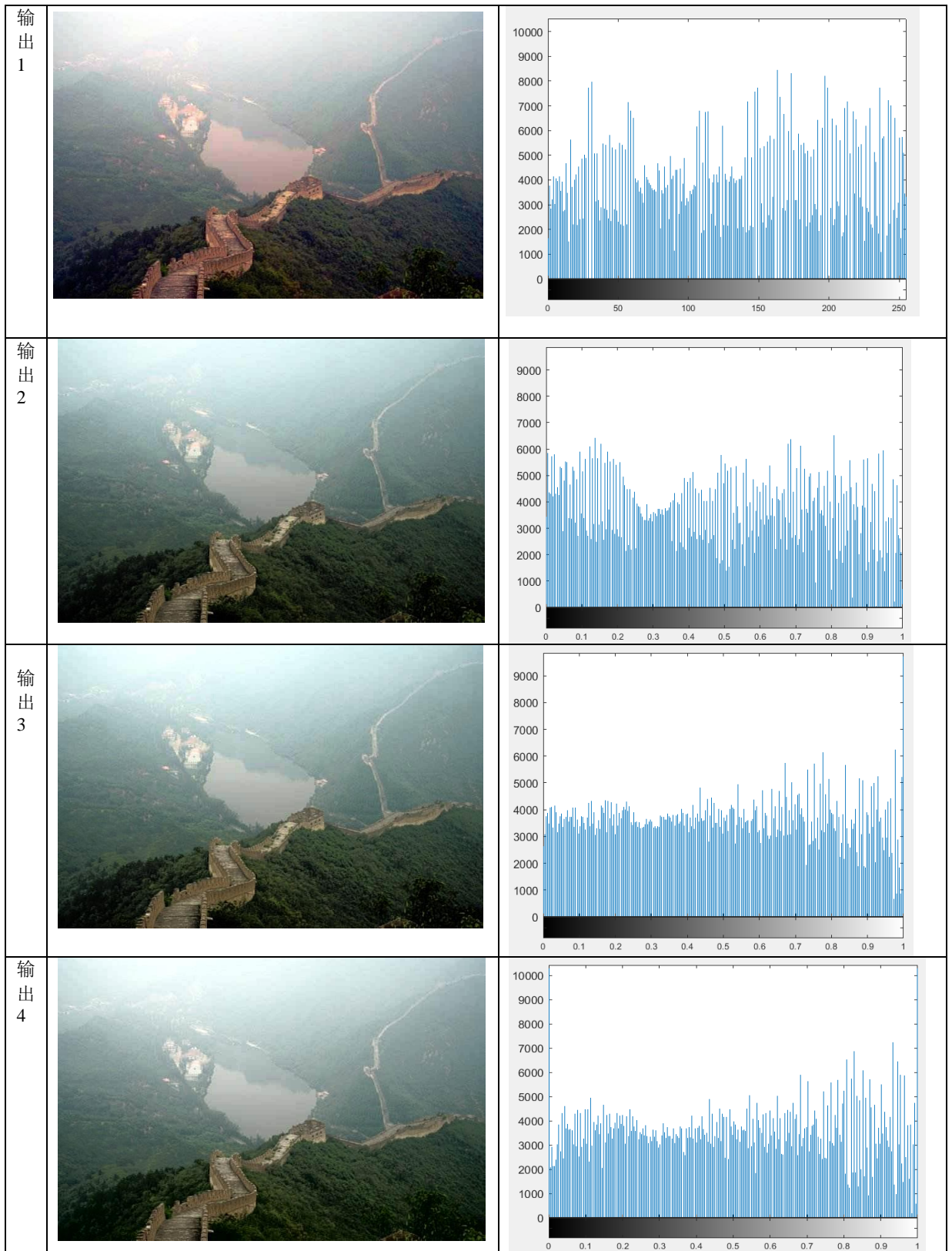
2.2 实验结果

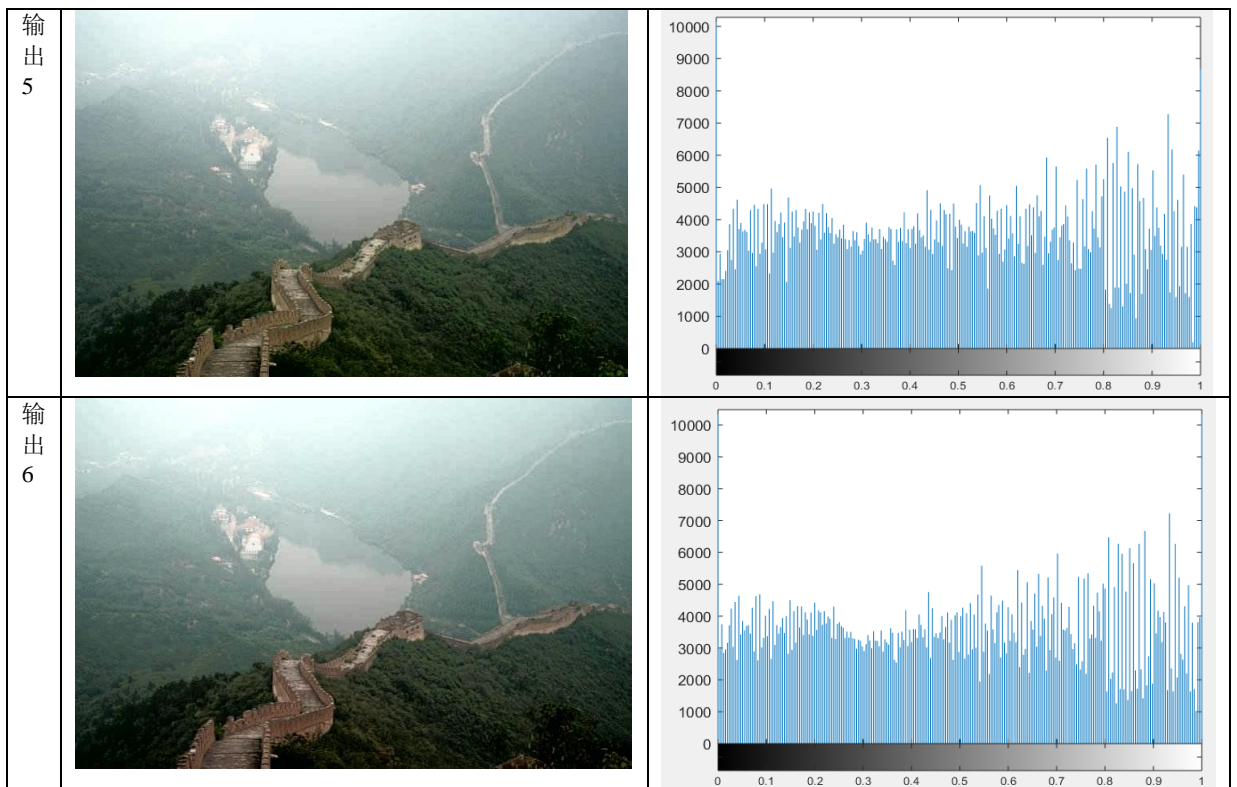
2.2.1 gray.jpg



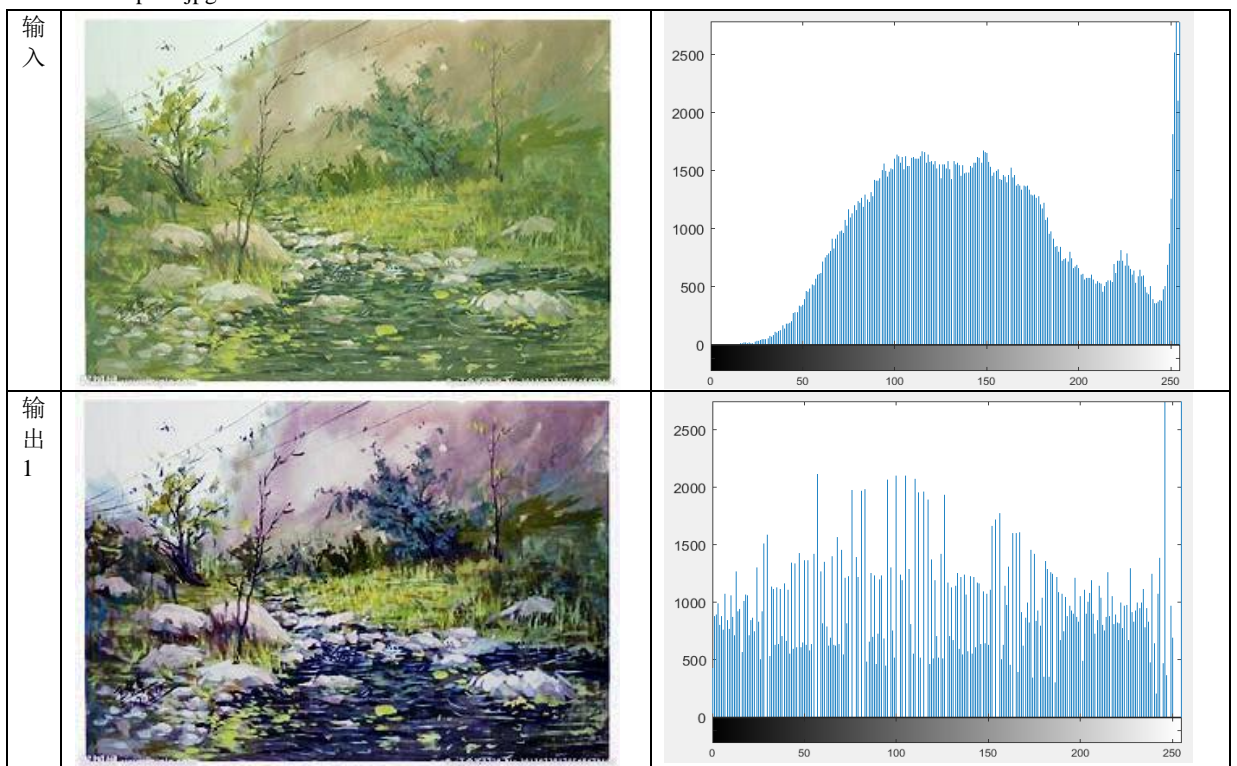
2.2.2 color.jpg

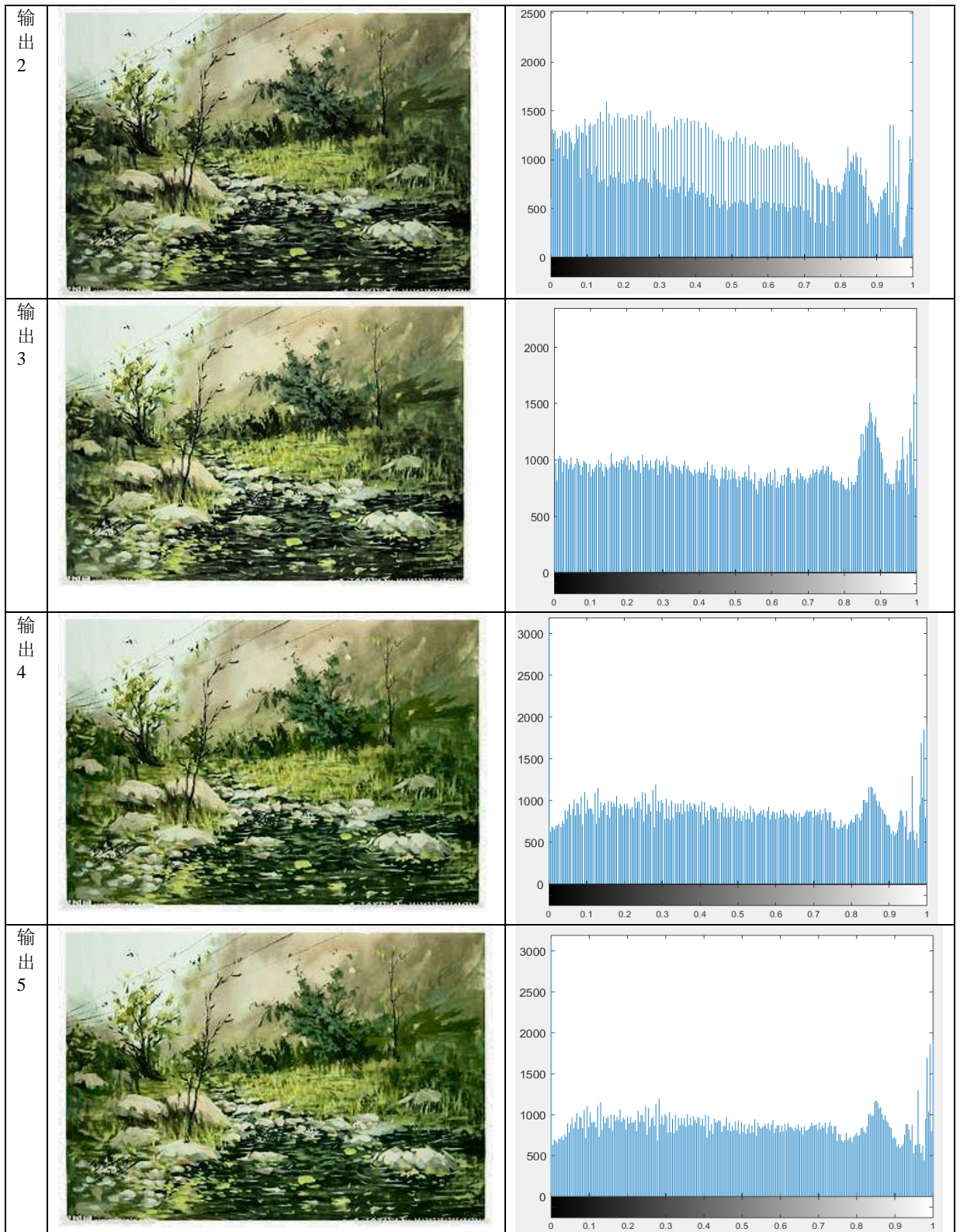


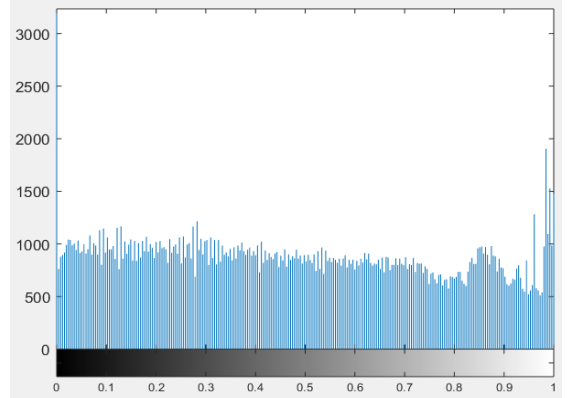




2.2.3 sample0.jpg

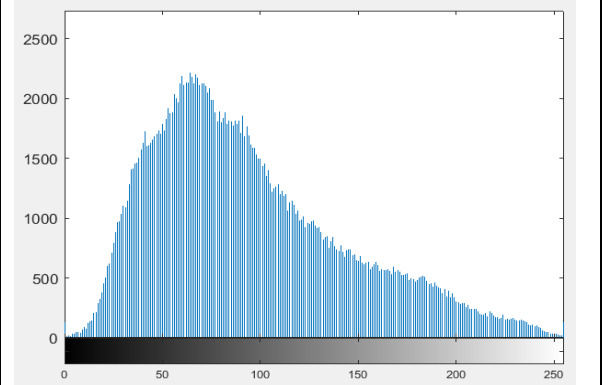
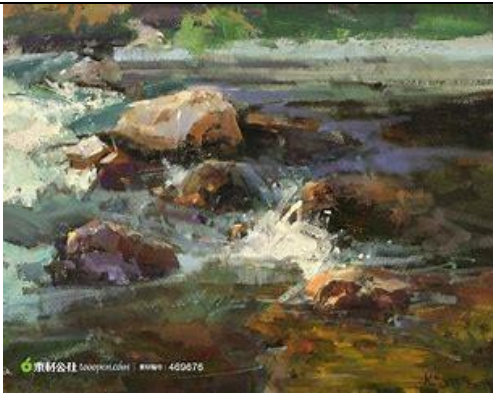
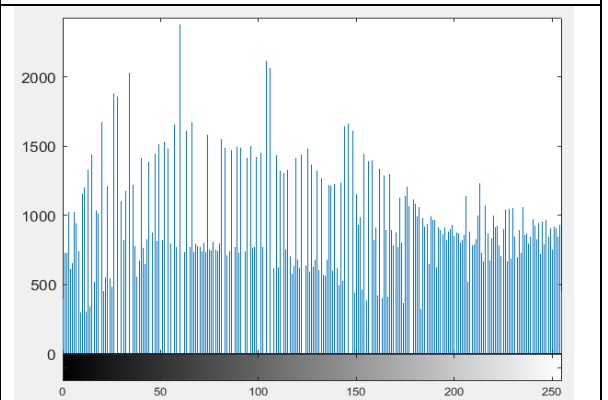



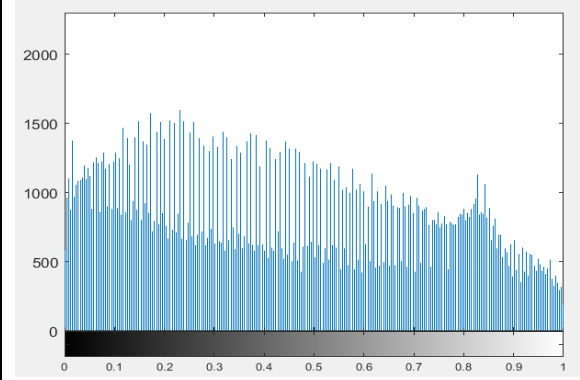

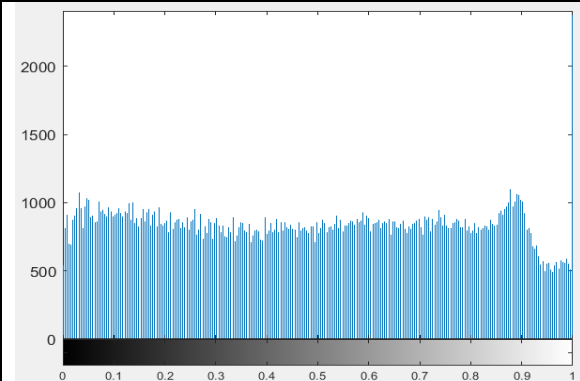
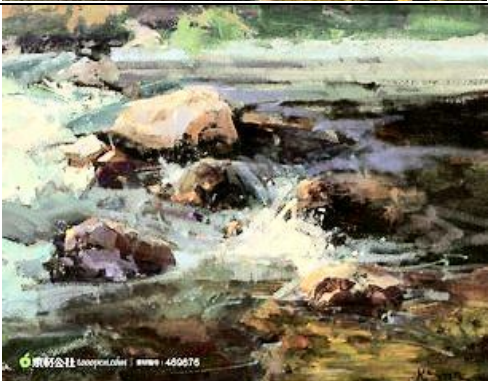
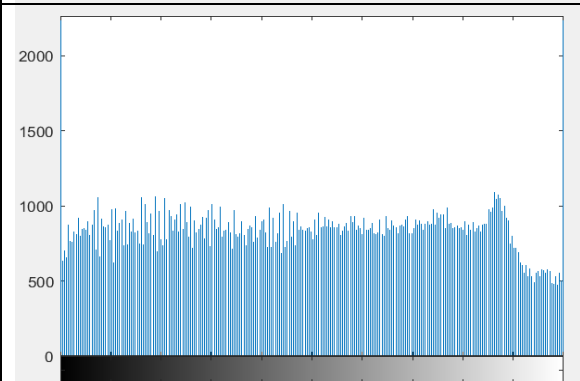
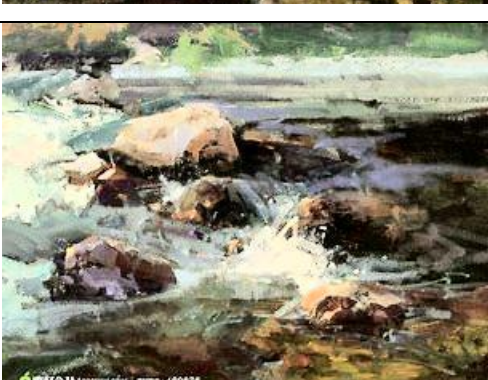
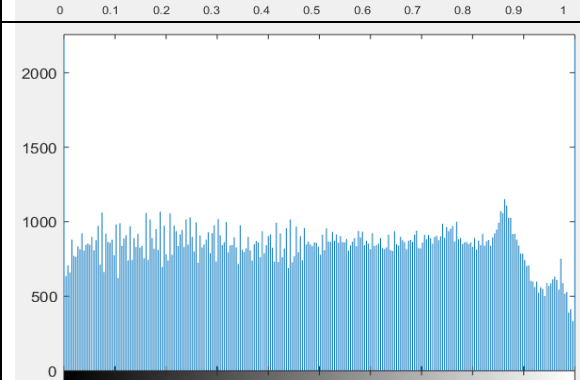


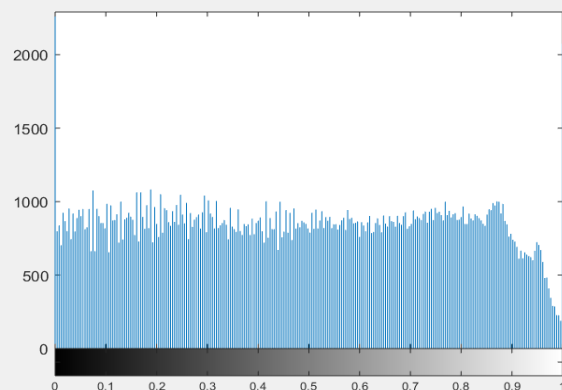
输出
6

2.2.4 sample1.jpg

输入

输出
1

输出 2		
输出 3		
输出 4		
输出 5		

输出
6

2.3 实验对比结论

综合图片效果和直方图的均衡度考虑,采用 RGB 转化为 HIS 之后再对 I 分量进行直方图均衡化得到增强后的图片效果最好。