

实验三 图像几何变换、彩色图像处理及压缩

课程名称：图像处理基础

实验学时：4 学时

综合性、设计性实验：☒是 ☒否

面向专业和班级：18 级软件工程（1 班、2 班、3 班、4 班、中澳班、卓越 1 班、卓越 2 班）

学生人数：102

任务编制人：课程负责人

实验内容

- 一、 图像几何变换
- 二、 彩色图像处理（1）
- 三、 彩色图像处理（2）
- 四、 图像压缩

一、图像的几何变换

(一) 实验目的

- 1.理解和掌握图像的平移、垂直镜像变换、水平镜像变换、缩放和旋转的原理和应用。

(二) 实验步骤

1. 对原图放大 2 倍。
2. 对原图进行旋转。
3. 对原图同时进行垂直镜像变换和水平镜像变换。
4. 对原图进行水平和垂直移动。

(三) 实验源码

- 1.对原图使用最近邻插值法、双线性插值和三次立方插值分别进行放大 2 倍(可以查看帮助文档找到其余的方法)

```
i=imread('flower.jpg')
j=imresize(i,2,'nearest')
figure(1),imshow(i)
figure(2),imshow(j)
```

- 2.对原图使用最近邻插值法、双线性插值和三次立方插值分别进行旋转

```
i=imread(' flower.jpg')
j=imrotate(i,45,'bilinear')
subplot(1,2,1),imshow(i);title('原图')
subplot(1,2,2),imshow(j);title('旋转 45 度')
```

- 3.对原图同时进行垂直镜像变换和水平镜像变换

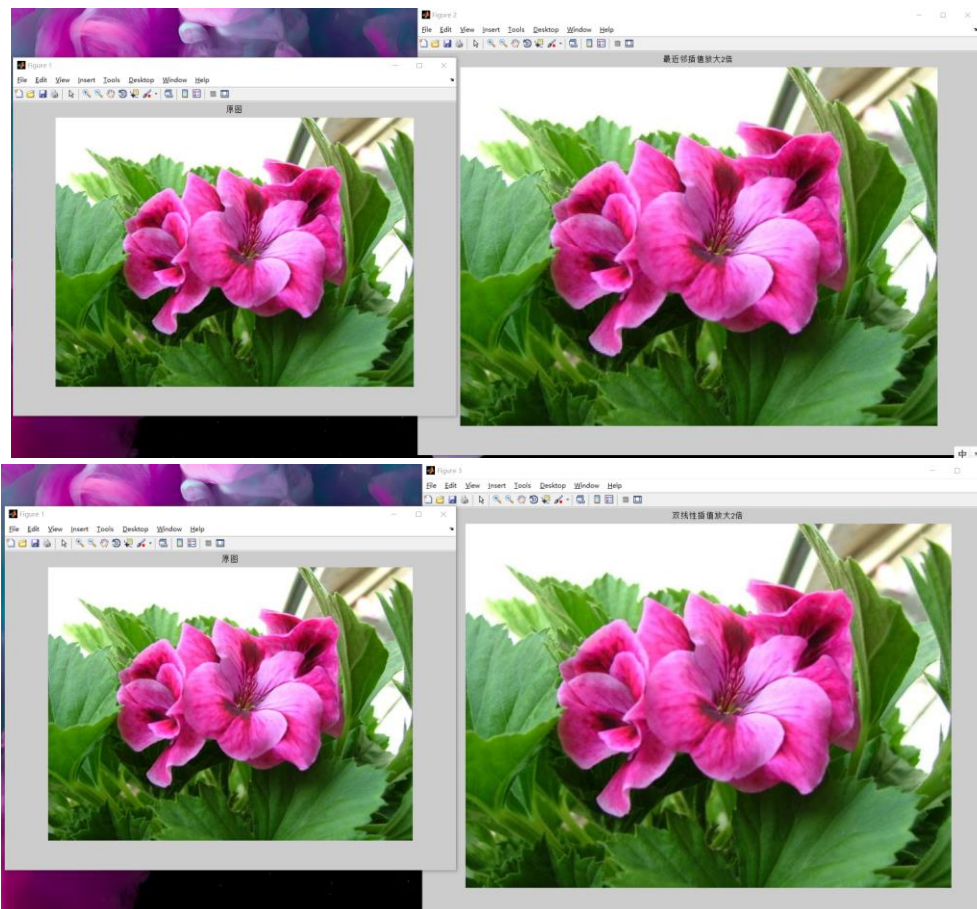
```
i=imread(' flower.jpg')
i=rgb2gray(i)
[x,y]=size(i)
p=1:x;q=1:y
m(x-p+1,y-q+1)=i(p,q)
subplot(1,2,1),imshow(i);title('原图')
subplot(1,2,2),imshow(m);title('先水平镜像后垂直镜像')
```

- 4.对原图像进行水平平移，垂直平移

代码自己写

(四) 实验截图

(1) 对原图使用最近邻插值法进行放大 2 倍



(2) 对原图使用双线性插值法进行旋转的处理如下：



(3) 对原图同时进行垂直镜像变换和水平镜像变换的处理结果:



(4) 对原图像进行水平平移，垂直平移:



二、彩色图像处理（1）

（一）实验目的

1、使用MatLab 软件对图像进行彩色处理，熟悉使用MatLab软件进行图像彩色处理的有关方法，并体会到图像彩色处理技术以及对图像处理的效果。

（二）实验要求

- 1、要求学生能够完成彩色图像的分析，能正确讨论彩色图像的亮度、色调等性质；
- 2、会对彩色图像进行直方图均衡，并能正确解释均衡处理后的结果；
- 3、能够对单色图像进行伪彩色处理、利用多波长图像进行假彩色合成、进行单色图像的彩色变换。

（三）实验内容与步骤

（1）彩色图像的分析

调入并显示彩色图像flower.jpg；

拆分这幅图像，并分别显示其R，G，B分量；

根据各个分量图像的情况讨论该彩色图像的亮度、色调等性质。

（2）彩色图像的直方图均衡——接内容（1）；

显示这幅图像的R，G，B分量的直方图，分别进行直方图均衡处理，并显示均衡后的直方图和直方图均衡处理后的各分量；

将处理完毕的各个分量合成彩色图像并显示其结果；

观察处理前后图像的彩色、亮度、色调等性质的变化。

（3）假彩色处理

调入并显示红色可见光的灰度图像v1_red.jpg、绿色可见光的灰度图像v1_green.jpg和蓝色可见光的灰度图像v1_blue.jpg；

以图像v1_red.jpg为R；图像v1_green.jpg为G；图像v1_blue.jpg为B，将这三幅图像组合成可见光RGB彩色图像；

（4）伪彩色处理1：灰度切片处理

调入并显示灰度图像lenna_gray.jpg；

利用MATLAB提供的函数对图像在8~256级的范围内进行切片处理，并使用hot模式和cool模式进行彩色化；

观察处理的结果。

（5）彩色变换(选做)

调入并显示灰度图像lenna_gray.jpg;

使用不同相位的正弦函数作为变换函数，将灰度图像变换为RGB图像。其中红色分量R的变换函数为 $-\sin()$ ，绿色分量G的变换函数为 $-\cos()$ ；蓝色分量B的变换函数为 $\sin()$ ；

显示变换曲线及变换合成的彩色图像并观察彩色变换图像的色调与原始图像灰度之间的关系；

将RGB的变换公式至少互换一次(例如R与G互换)，显示变换曲线、变换结果并观察处理的结果。

(6) 打印全部结果并进行讨论。

利用MATLAB软件实现彩色图像处理的程序：

```
rgb_image=imread('flower.jpg'); %读取图像 flower.jpg
fR=rgb_image(:,:,1);           %获取图像的红色分量
fG=rgb_image(:,:,2);           %获取图像的绿色分量
fB=rgb_image(:,:,3);           %获取图像的蓝色分量
figure(1),imshow(fR)            %分别显示图像
figure(2),imshow(fG)
figure(3),imshow(fB)

%实现 rgb 图像转化为 NTSC 彩色空间的图像
yiq_image=rgb2ntsc(rgb_image);
fY=yiq_image(:,:,1);           %图像 flower.jpg 的亮度
fI=yiq_image(:,:,2);           %图像 flower.jpg 的色调
fQ=yiq_image(:,:,3);           %图像 flower.jpg 的饱和度
figure(4),imshow(fY)
figure(5),imshow(fI)
figure(6),imshow(fQ)

fR=histeq(fR,256);             %对彩色图像的分量进行直方图均衡化
fG=histeq(fG,256);
fB=histeq(fB,256);
RGB_image=cat(3,fR,fG,fB); %将直方图均衡化后的彩色图像合并
figure,imshow(RGB_image) %观察处理后的彩色图色度，亮度参照前面
f1=imread('v1_red.jpg');
f2=imread('v1_green.jpg');
f3=imread('v1_blue.jpg');
f4=imread('infer_near.jpg');
ture_color=cat(3,f1,f2,f3);
figure,imshow(ture_color) %显示由红、绿、蓝三幅图合成的彩色图

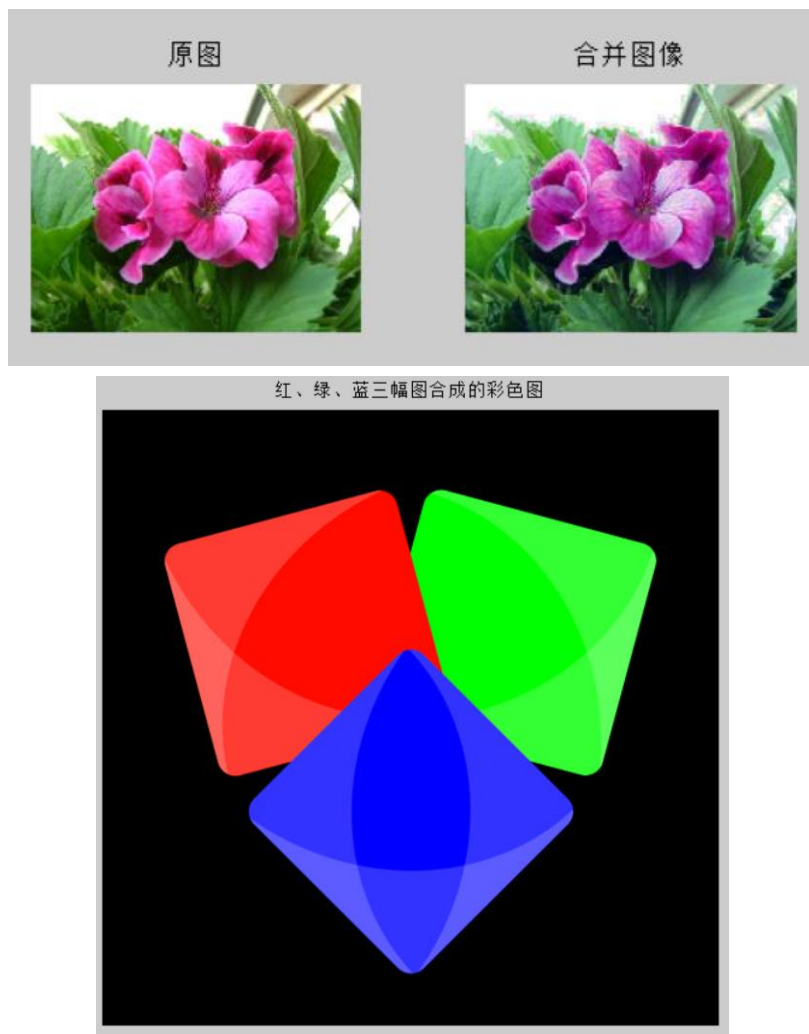
f=imread('head.jpg');
cut_1=imadjust(f,[0.0925 0.5],[0.0925 0.5]);%提取灰度在 16-128 之间的像素
cut_2=imadjust(f,[0.5 1],[0.5 1]); %提取灰度在 128-256 之间的像素
figure,imshow(cut_1),colormap(hot) %显示图像 cut_1,并使用 hot 模型彩色化
```

```
figure,imshow(cut_2,colormap(cool))%显示图像 cut_2,并使用 cool 模型彩色化  
(选做)  
f=imread('Lenna.jpg');  
g=ice('image',f);           %通过 ice(交互彩色编辑)函数对图像进行变换  
ice(交互彩色编辑)函数的参数参照书后面的注释。
```

(四) 思考题

1. 为什么经彩色直方图均衡后的图像除了对比度会有所增强外，还有色调的变化？
2. 实验内容 (3) 的假彩色处理方案是否可以有多种？若有，请估计其它方案的可能结果。
3. 在实验内容 (4) 中，对于灰度切片处理的图像 head.gif 使用多少级切片比较合适？

(五) 部分实验效果



三、MATLAB 实现彩色图像处理（2）

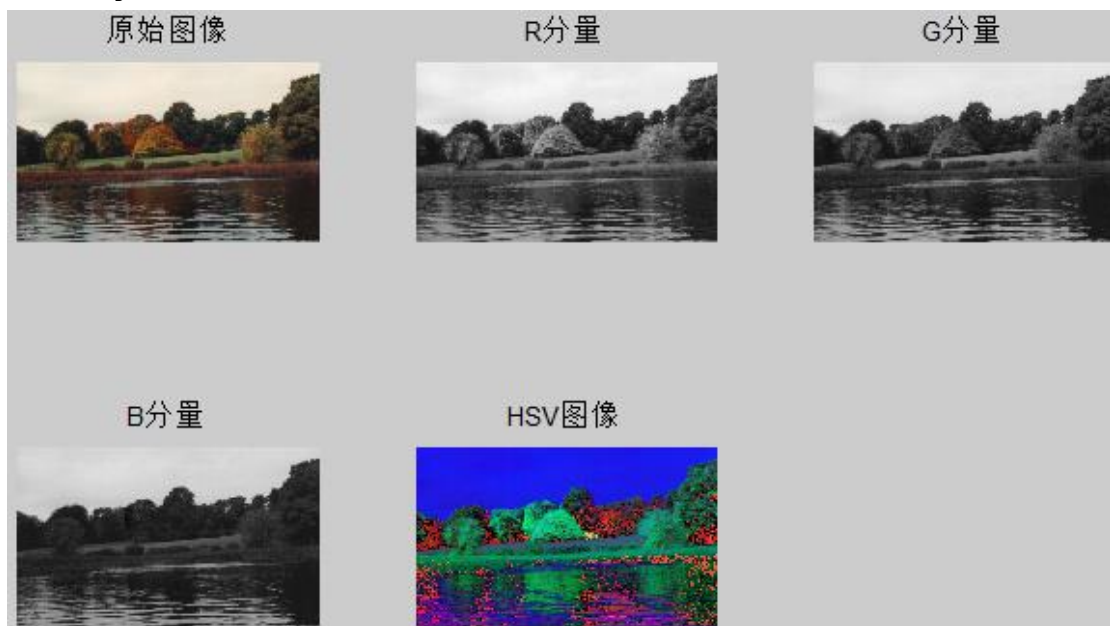
（一）实验内容

1、 任选一幅彩色图像(RGB)

(1)彩色图像的分析

调入并显示彩色图像；拆分这幅图像，并分别显示其 R，G，B 分量；将该图像转换成 HSV 图像，根据各个分量图像的情况讨论该彩色图像的亮度、色调等性质。

```
RGB=imread('autumn.tif');  
R=RGB(:,:,1);  
G=RGB(:,:,2);  
B=RGB(:,:,3);  
hsv=rgb2hsv(RGB);
```



(2) 彩色图像的直方图均衡

显示这幅图像的 R，G，B 分量的直方图，分别进行直方图均衡处理，并显示均衡后的直方图和直方图均衡处理后的各分量；将处理完毕的各个分量合成彩色图像并显示其结果；将该图像转换成 HSV 图像，观察处理前后图像的彩色、亮度、色调等性质的变化。

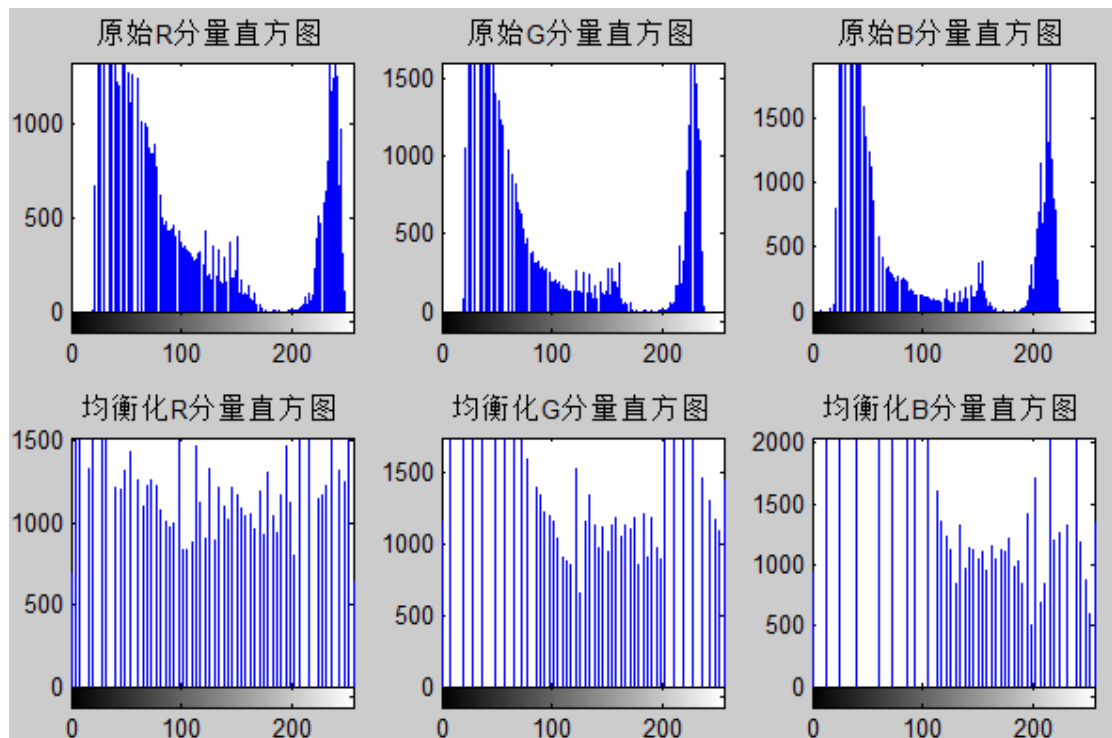
```
RGB=imread('autumn.tif');  
hsv=rgb2hsv(RGB);  
R=RGB(:,:,1);subplot(231),imhist(R),title('原始 R 分量直方图');  
G=RGB(:,:,2);subplot(232),imhist(G),title('原始 G 分量直方图');  
B=RGB(:,:,3);subplot(233),imhist(B),title('原始 B 分量直方图');
```

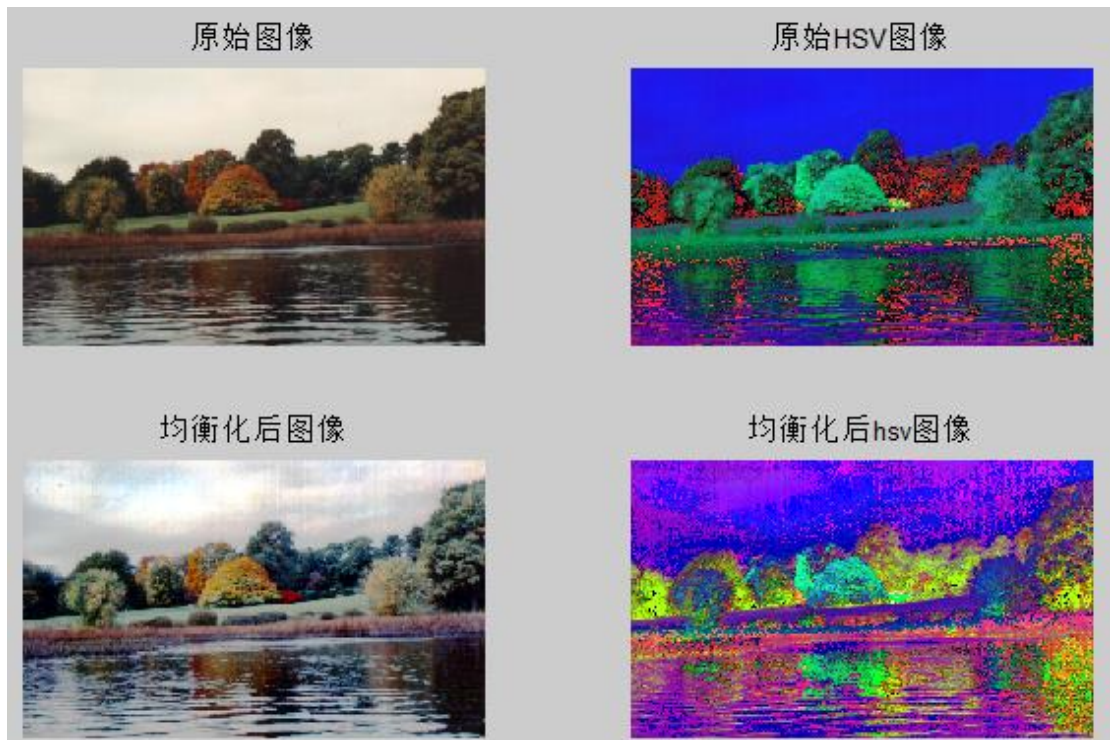


```

R1=histeq(R);subplot(234),imhist(R1),title('均衡化 R 分量直方图');
G1=histeq(G);subplot(235),imhist(G1),title('均衡化 G 分量直方图');
B1=histeq(B);subplot(236),imhist(B1),title('均衡化 B 分量直方图');
RGB1=cat(3,R1,G1,B1);
hsv1=rgb2hsv(RGB1);
figure
subplot(221),imshow(RGB),title('原始图像');
subplot(222),imshow(hsv),title('原始 HSV 图像');
subplot(223),imshow(RGB1),title('均衡化后图像');
subplot(224),imshow(hsv1),title('均衡化后 hsv 图像');

```





(3) 彩色图像在 HSI 模型下的增强(HSI 和 HSV 图像有区别, 可自行比较)

将一幅 RGB 彩色图像转换为 HSI 空间。分别对 H, S, I 三个分量进行增强, 显示结果, 并分析

```

RGB=imread('autumn.tif');
[hsi,H,S,I]=rgb2hsi(RGB);
subplot(3,4,1),imshow(RGB),title('原始图像');
subplot(3,4,2),imshow(H),title('H 分量');
subplot(3,4,3),imshow(S),title('s 分量');
subplot(3,4,4),imshow(I),title('I 分量');
subplot(3,4,5),imshow(hsi),title('hsi 图像');
h=fspecial('prewitt');
H1=imfilter(H,h);
S1=imfilter(S,h);
I1=imfilter(I,h);
subplot(3,4,6),imshow(H1),title('H 分量锐化');
subplot(3,4,7),imshow(S1),title('s 分量锐化');
subplot(3,4,8),imshow(I1),title('I 分量锐化');
hsi1=cat(3,H1,S1,I1);
subplot(3,4,9),imshow(hsi1),title('新 hsi 图像');
RGB1=hsi2rgb(hsi1);
subplot(3,4,10),imshow(RGB1),title('新 RGB 图像');

```



2、伪彩色处理

(1)、灰度分层法处理

调入并显示灰度图像；利用 MATLAB 提供的函数 `grayslice` 对图像在 8~256 级的范围内进行切片处理，并使用 `hot` 模式和 `cool` 模式进行彩色化；

```
RGB=imread('lenna_color.jpg');
I=rgb2gray(RGB);
J=grayslice(I,8);
subplot(131),imshow(RGB),title('原始彩色图像');
subplot(132),imshow(I),title('原始灰度图像');
subplot(133),imshow(J,[],),title('8 切片图像');
figure
subplot(121),imshow(RGB),title('原始彩色图像');
subplot(122),imshow(J,hot(8)),title('hot 恢复彩色图像');
figure,imshow(J,cool(8)),title('cool 恢复彩色图像');
```

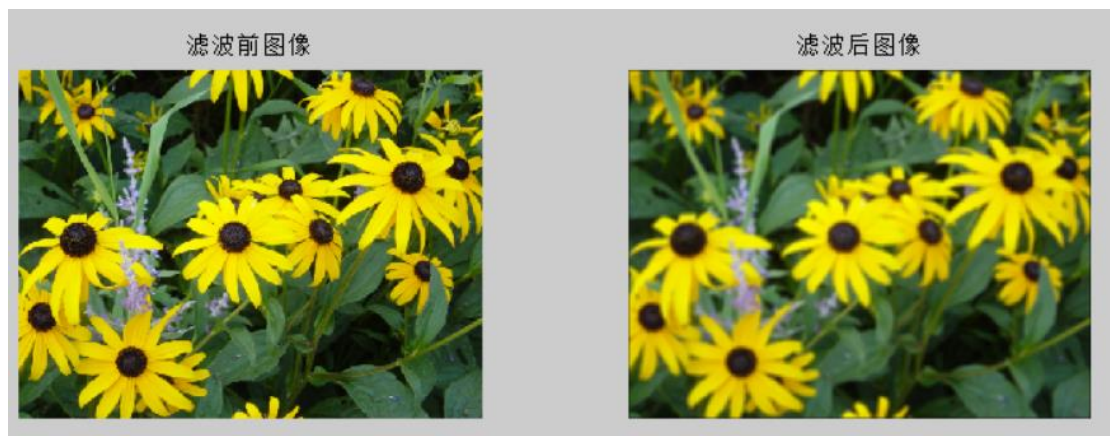




(2) 编程序实现其他两种伪彩色增强法，比较显示结果。(选作)

3、运行如下程序，体验彩色图像平滑，分析前后变化：

```
RGB=imread('flowers.tif');
H=ones(5,5)/25;
RGB1=imfilter(RGB,H);
subplot(1,2,1),imshow(RGB),title('滤波前图像')
subplot(1,2,2),imshow(RGB1),title('滤波后图像')
```



4、运行如下程序，体验假彩色图像处理，分析前后变化：

```
clc;[RGB]=imread( 'autumn.tif' );imshow(RGB),title( '原始图像');  
RGBnew(:,:,1)=RGB(:,:,3);RGBnew(:,:,2)=RGB(:,:,1);  
RGBnew(:,:,3)=RGB(:,:,2);figure,imshow(RGBnew),title( '假彩色变换后图像' );
```



(二) 思考题

- 1、伪彩色增强和假彩色增强的差别在哪里？
- 2、彩色图像增强在 RGB 空间进行，和在 HIS 空间进行有何差别？

四、图像压缩

(一) 实验目的

1. 理解有损压缩和无损压缩的概念；
2. 理解图像压缩的主要原则和目的；
3. 了解几种常用的图像压缩编码方式。
4. 利用 MATLAB 程序进行图像压缩。

(二) 实验原理

1. 图像压缩原理

图像压缩主要目的是为了节省存储空间，增加传输速度。图像压缩的理想标准是信息丢失最少，压缩比例最大。不损失图像质量的压缩称为无损压缩，无损压缩不可能达到很高的压缩比；损失图像质量的压缩称为有损压缩，高的压缩比是以牺牲图像质量为代价的。压缩的实现方法是对图像重新进行编码，希望用更少的数据表示图像。

信息的冗余量有许多种，如空间冗余，时间冗余，结构冗余，知识冗余，视觉冗余等，数据压缩实质上是减少这些冗余量。高效编码的主要方法是尽可能去除图像中的冗余成分，从而以最小的码元包含最大的图像信息。

编码压缩方法有许多种，从不同的角度出发有不同的分类方法，从信息论角度出发可分为两大类。

(1) 冗余度压缩方法，也称无损压缩、信息保持编码或熵编码。具体说就是解码图像和压缩编码前的图像严格相同，没有失真，从数学上讲是一种可逆运算。

(2) 信息量压缩方法，也称有损压缩、失真度编码或熵压缩编码。也就是说解码图像和原始图像是有差别的，允许有一定的失真。

应用在多媒体中的图像压缩编码方法，从压缩编码算法原理上可以分为以下 3 类：

(1) 无损压缩编码种类

哈夫曼 (Huffman) 编码，算术编码，行程 (RLE) 编码，Lempel zev 编码。

(2) 有损压缩编码种类

预测编码，DPCM，运动补偿；

频率域方法：正交变换编码(如 DCT)，子带编码；

空间域方法：统计分块编码；

模型方法：分形编码，模型基编码；

基于重要性：滤波，子采样，比特分配，向量量化；

(3) 混合编码。

有 JBIG，H261，JPEG，MPEG 等技术标准。

本实验主要利用 MATLAB 程序进行离散余弦变换 (DCT) 压缩和行程编码 (Run Length Encoding, RLE)。

1) 离散余弦变换(DCT)图像压缩原理

离散余弦变换 DCT 在图像压缩中具有广泛的应用，它是 JPEG、MPEG 等数据压缩标准

的重要数学基础。

和相同图像质量的其他常用文件格式(如 GIF(可交换的图像文件格式), TIFF(标签图像文件格式), PCX(图形文件格式))相比, JPEG 是目前静态图像中压缩比最高的。JPEG 比其他几种压缩比要高得多,而图像质量都差不多(JPEG 处理的图像只有真彩图和灰度图)。正是由于其高压缩比,使得 JPEG 被广泛地应用于多媒体和网络程序中。JPEG 有几种模式,其中最常用的是基于 DCT 变换的顺序型模式,又称为基本系统(Baseline)。

2) 用 DCT 压缩图像的过程为:

(1) 首先将输入图像分解为 8×8 或 16×16 的块,然后对每个子块进行二维 DCT 变换。

(2) 将变换后得到的量化的 DCT 系数进行编码和传送,形成压缩后的图像格式。

3) 用 DCT 解压的过程为:

(1) 对每个 8×8 或 16×16 块进行二维 DCT 反变换。

(2) 将反变换的矩阵的块合成一个单一的图像。

余弦变换具有把高度相关数据能量集中的趋势,DCT 变换后矩阵的能量集中在矩阵的左上角,右下的大多数的 DCT 系数值非常接近于 0。对于通常的图像来说,舍弃这些接近于 0 的 DCT 的系数值,并不会对重构图像的画面质量带来显著的下降。所以,利用 DCT 变换进行图像压缩可以节约大量的存储空间。压缩应该在最合理地近似原图像的情况下使用最少的系数。使用系数的多少也决定了压缩比的大小。

在压缩过程的第 2 步中,可以合理地舍弃一些系数,从而得到压缩的目的。在压缩过程的第 2 步,还可以采用 Huffman 编码来进一步压缩。

(三) 实验程序与结果

(1) 8×8 :

```
I=imread('lena.jpg');
K=rgb2gray(I);
I=im2double(K);
imshow(I);
T=dctmtx(8);
B=blkproc(I,[8 8], 'P1*x*P2',T,T);
mask=[1 1 1 1 0 0 0 0
      1 1 1 0 0 0 0 0
      1 1 0 0 0 0 0 0
      1 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0];
B2=blkproc(B,[8 8], 'P1.*x',mask);
```

```

I2=blkproc(B2,[8 8], 'P1*x*P2',T',T);
I=imread('lena.jpg');
subplot(2,2,1);
imshow(I);
title('原始图像');
subplot(2,2,2);
imshow(K);
title('灰度图');
subplot(2,2,3);
imshow(B2);
title('压缩图');
subplot(2,2,4);
imshow(I2);
title('复原图');

```

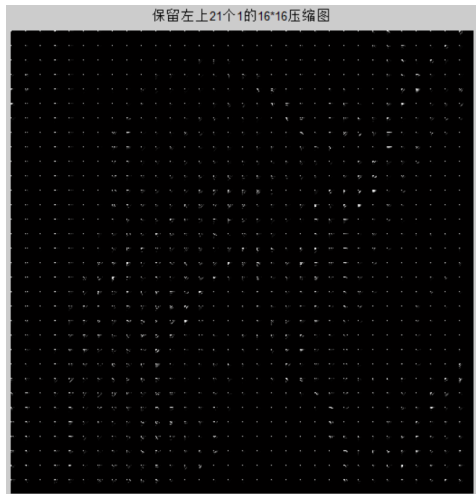
效果图:



(2)16*16:

代码自己写

效果图：



(3)实验结果分析

对比分析两次实验结果。

(四) 思考题

1、简述离散余弦变换（DCT）的原理。