

完成时间：2022.12.06

视频编码与通信 第一次作业报告

2022 年

作业名称：灰度色度大作业

姓名：冯绍庭

学号：520021911362



目 录

1. 实验目的	1
2. 量化灰度	1
2.1 基本概念	1
2.2 实验要求	1
2.3 实验思路	1
2.4 实验结果与分析	1
3. YCrCb 彩色图片采样	3
3.1 基本概念	3
3.2 实验要求	4
3.3 实验思路	4
3.4 实验结果与分析	4
4. DCT 变换	5
4.1 基本概念	5
4.2 实验要求	5
4.3 实验思路	5
4.4 实验结果与分析	6
4.4.1 2D-DCT 变换与反变换后的图像	6
4.4.2 图像的频谱图	8
5. 附录 软件程序清单	8
5.1 量化灰度和 YCrCb 彩色图片采样	8
5.2 DCT 变换	11

1. 实验目的

1. 理解不同量化层次对图像质量的影响
2. 观察人眼对灰度和色度的敏感程度的不同
3. 理解 DCT 在图像压缩中的作用

2. 量化灰度

2.1 基本概念

量化分层指的是对图像采样后的采样值（模拟量）进行离散化。量化越细，量化阶数（灰度级）越多，量化误差就越小，但数字图像的数据量就越大。人眼对灰度误差有一个敏感阈值，当灰度误差小于阈值时，人眼是无法感知的，所以量化层次多到一定程度，量化误差就可以小于视觉阈值。如果量化误差过大，看图时就会出现伪轮廓。伪轮廓属于数字图像噪声中的一种，主要表现为在数字图像或者视频信号中所呈现的画面平场区域出现明显的类似于等高线的不连续过渡带，从而对于画面感观质量造成影响的现象。

2.2 实验要求

将一张图片分别量化为 2 个灰度级、4 个灰度级、8 个灰度级、64 个灰度级、128 个灰度级和 256 个灰度级，分别给出相应的灰度图和彩色图。

2.3 实验思路

Matlab 中用 `imread`（）函数读取图片后存储在三维数组 `img_org_data` 中，三个维度的分别对应图片的长、宽、像素点的 RGB 参数。由于原图的量化阶数为 256，设目标量化阶为 L ，则原始图片中每 $d = 255/(L - 1)$ 个量化层需要合并为一个量化层。代码中我们需要将原始图片数组 `img_org_data` 中每个数值除以 d 四舍五入后乘 d 再四舍五入，便可得到量化后的彩色图。若要得到灰度图只需要对前两个维度进行上述处理。

2.4 实验结果与分析

图片原图如图 2.1 所示，不同灰度级的灰度图和彩色图如图 2.2 所示。

从图 2.2 可以看出，随着量化级数的减少，图像变得越来越模糊，灰色部分逐渐被纯黑色和白色所取代。由于量化误差的增加，图像质量变差，在级数小于 64 级后，出现了明显的伪轮廓。

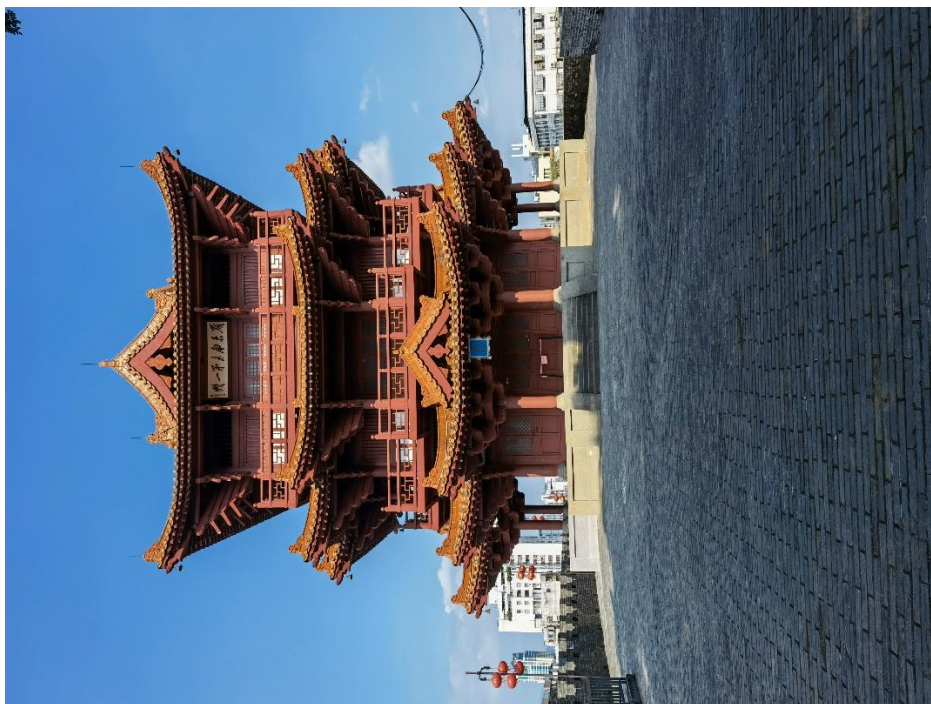
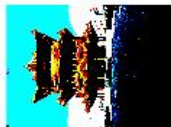


图 2.1 图片原图

2个灰度级（彩色图）



4个灰度级（彩色图）



8个灰度级（彩色图）



64个灰度级（彩色图）



128个灰度级（彩色图）



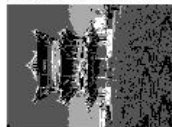
256个灰度级（彩色图）



2个灰度级（灰度图）



4个灰度级（灰度图）



8个灰度级（灰度图）



64个灰度级（灰度图）



128个灰度级（灰度图）



256个灰度级（灰度图）



图 2.2 不同灰度级的灰度图和彩色图

3. YCrCb 彩色图片采样

3.1 基本概念

RGB 要求三个独立的视频信号同时传输，相比之下，YCrCb 可以优化彩色视频信号的传输，只需占用极少的带宽。Y 表示**亮度**，由 RGB 信号的特定部分叠加到一起而建立。Cr 和 Cb 表示**色度**，定义了颜色的两个方面——色调与饱和度。Cr 反映了 RGB 输入信号红色部分与 RGB 信号亮度值之间的差异。而 Cb 反映的是 RGB 输入信号蓝色部分与 RGB 信号亮度值之间的差异。

RGB 与 YCrCb 的转化方程如下：

$$Y = 0.257 * R + 0.504 * G + 0.098 * B + 16 \tag{3.1}$$

$$Cb = -0.148 * R - 0.291 * G + 0.439 * B + 128 \tag{3.2}$$

$$Cr = 0.439 * R - 0.368 * G - 0.071 * B + 128 \tag{3.3}$$

根据 Y:Cr:Cb 的值不同，常见的 YCrCb 的采样可以分为 4:4:4、4:2:2、4:4:1、4:2:0 四类。采样的具体过程由图 3.1 所示。

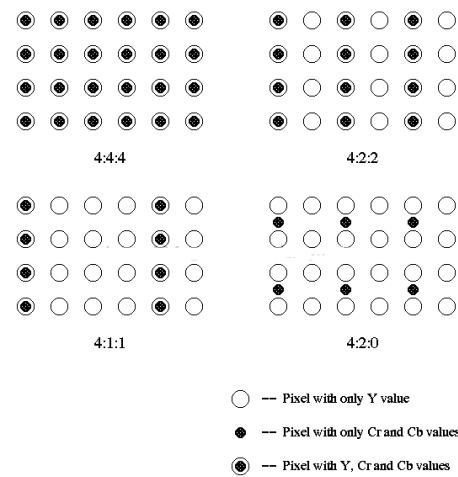


图 3.1 采样的具体过程

4:2:0 这种采样的 0 指 Cb、Cr 分量隔行才采样一次。比如第一行采样 4:2:0 ,第二行采样 4:0:2 ,依次类推.....在这种采样方式下，每四个 Y 共用一组 Cb、Cr 分量。下采样各分量保存百分比如表 3.1 所示。

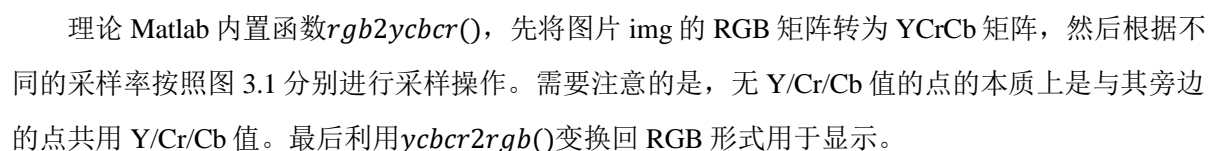
表 3.1 下采样各分量保存百分比

采样率	Y 保存百分比	Cr、Cb 保存百分比
4:4:4	1	0.5
4:2:2	1	0.5
4:1:1	1	0.25
4:2:0	1	0.25

3.2 实验要求

选择一个 4:4:4 格式的彩色图片，下采样为 4:2:2、4:1:1 和 4:2:0 几种不同的格式，分别给出相应的彩色图片，观察并简单分析结果。

3.3 实验思路

理论 Matlab 内置函数`rgb2ycbcr()`，先将图片的 RGB 矩阵转为 YCbCr 矩阵，然后根据不同的采样率按照图 3.1 分别进行采样操作。需要注意的是，无 Y/Cr/Cb 值的点的本质上是与其旁边的点共用 Y/Cr/Cb 值。最后利用`ycbcr2rgb()`变换回 RGB 形式用于显示。

3.4 实验结果与分析

分别在四种采样率下采样，再变换回 RGB 形式，得到彩色图片如图 3.2 所示。观察图 3.2，基本很难用肉眼察觉到色彩的失真，仔细观察才能够发现各图的颜色饱和度均有所降低。图 2.2 中的彩色图我们能够清晰地看出失真，可以说明人眼对灰度更加敏感。但这需要辩证看待，因为采用不同量化级和不同采样方式，得到的结论会不同。



图 3.2 下采样

分别输出 Y、Cb、Cr 矩阵对应的图片，如图 3.3 所示，可以发现 Y 分量对应的图片已经包含了绝大部分原图片的信息，说明人眼对色度信号的分辨率比亮度信号低。

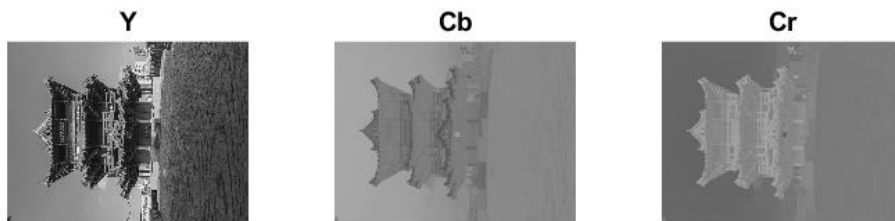


图 3.3 Y、Cb、Cr 矩阵对应的图片

4. DCT 变换

4.1 基本概念

DCT 全称离散余弦变换，经常用于信号和图像数据的压缩。经过 DCT 变换后的数据能量非常集中，一般只有左上角的数值是非零的，也就是能量都集中在离散余弦变换后的直流和低频部分。

DCT 公式如下：

$$F(u, v) = c(u)c(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j) \cos\left[\frac{(i+0.5)\pi}{N} u\right] \cos\left[\frac{(j+0.5)\pi}{N} v\right] \quad (4.1)$$

$$c(u) = \begin{cases} \sqrt{\frac{1}{N}}, & u = 0 \\ \sqrt{\frac{2}{N}}, & u \neq 0 \end{cases} \quad (4.2)$$

$$c(v) = \begin{cases} \sqrt{\frac{1}{N}}, & v = 0 \\ \sqrt{\frac{2}{N}}, & v \neq 0 \end{cases} \quad (4.3)$$

4.2 实验要求

选择一张图片，将其按 8x8 分块，对每一块分别作 8x8 的 2D-DCT 变换，并保留左上角前 2 条、前 4 条、前 6 条对角线上的系数（其余置 0）后作 8x8 的反变换，比较得到的图像与原图像并分析。

4.3 实验思路

由式(4.1)-(4.3)，一般只讨论两个 N 相等的情况，也就是数据是方阵的形式。由于我们对矩阵按 8 × 8 分块，所以对每一块如下变换：

$$F = AfA^T \quad (4.4)$$

$$A(i, j) = c(i) \cos\left[\frac{(j+0.5)\pi}{N} i\right] \quad (4.5)$$

我们可以用 Matlab 中的 *dctmtx*(8) 函数生成 8x8 的 DCT 系数矩阵 T ，再用 *blockproc* ($R, [8\ 8], fundct$) 函数进行 8x8 分块操作，其中的句柄函数 *fundct* 定义为 *fundct* = @(block_struct) $T * blockstruct.data * T'$ ，即分别用 DCT 矩阵乘以对应的 RGB 矩阵。再利用遮罩矩阵 *mask* 分别并保留左上角前 2 条、前 4 条、前 6 条对角线上的系数（其余置 0）后按照上述类似方法再进行 8x8 的反变换。

4.4 实验结果与分析

4.4.1 2D-DCT 变换与反变换后的图像

我们选择图 2.1 作为原始图像，将保留左上角前 2 条、前 4 条、前 6 条对角线上的系数（其余置 0）后作 8x8 的反变换得到的图像与原图进行对比，如图 4.1 所示。我们发现每一张图片都很清晰，这说明 DCT 能够在保留图片大部分信息的情况下，对图片进行压缩。我们更换图片来观察不同图片之间的差异。

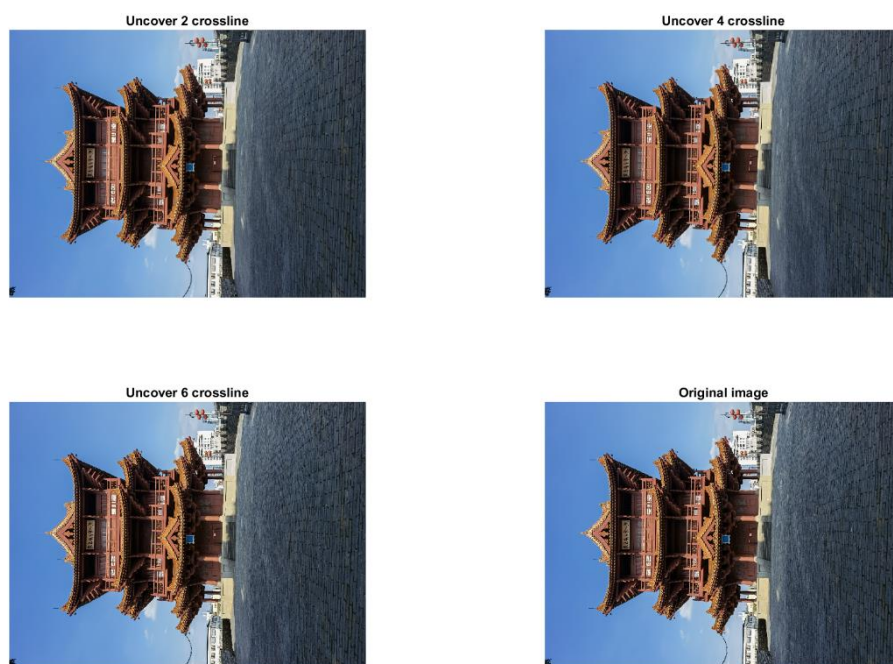


图 4.1 2D-DCT 变换与反变换后的图像

我们采用图像处理领域最经典 Lena 图中的主角 Lena 的另一张图片，如图 4.2 所示。观察图 4.3，这次显然保留对角线越多图像越清晰，并且对于仅保留直流分量的图片也能大概看出人物与物品的轮廓。



图 4.2 图片原图 (Lena)



图 4.3 2D-DCT 变换与反变换后的图像 (Lena)

4.4.2 图像的频谱图

对图 2.1 的二维灰度图像进行 DCT 变换，就能得到图像的频谱图：低阶（变化幅度小）的部分反映在 DCT 的左上方，高阶（变化幅度大）的部分反映在 DCT 的右下方。由于人眼对高阶部分不敏感，依靠低阶部分就能基本识别出图像内容。如图 4.4 所示，大部分 DCT 正系数集中在矩阵的左上角区域。

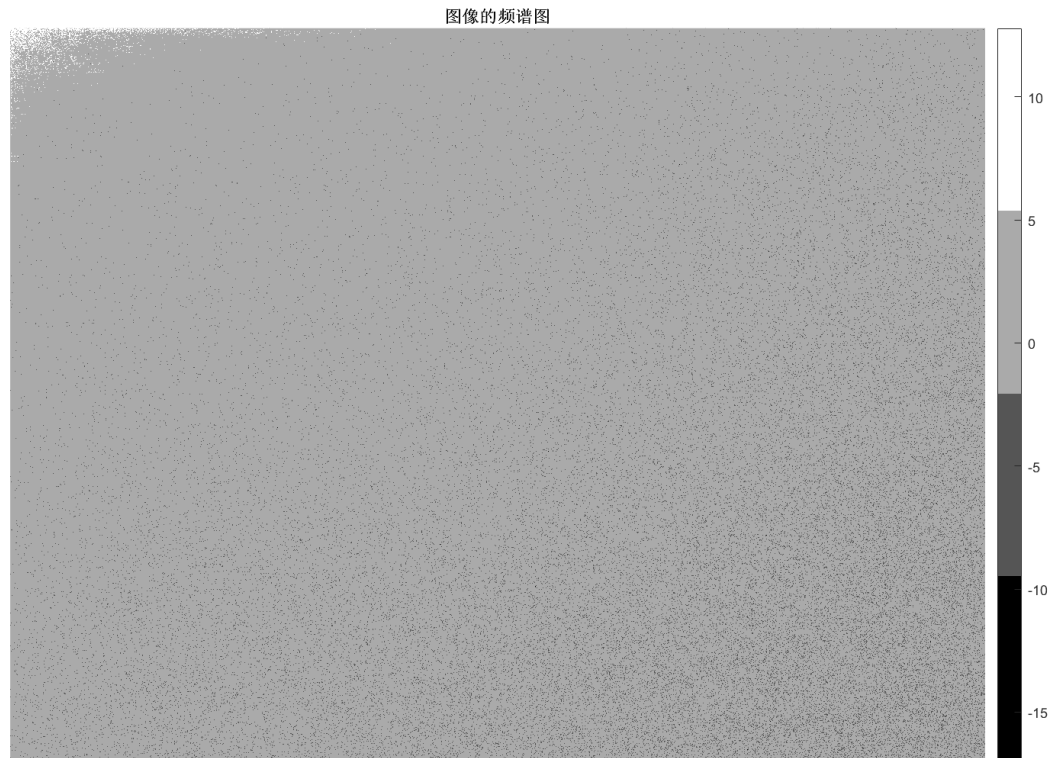


图 4.4 图像的频谱图

5. 附录 软件程序清单

5.1 量化灰度和 YCrCb 彩色图片采样

```
1. clear;
2. close all;
3.
4. %% Set Parameter
5. img_org = "img_org.jpg";
6. level = [2 4 8 64 128 256];
7.
8. %% 获取图片信息
```

```

9. img_data = imread(img_org);
10. % 旋转图片
11. for P = 1:3
12.     img_org_data(:, :, P) = img_data(:, :, P)';
13. end
14. [h,w,p] = size(img_org_data);
15.
16. figure(1);
17. imshow(img_org_data);
18.
19. %% 处理灰度级
20. figure(2);
21. for j = 1:6
22.     L = level(j);
23.     d = 255 / (L - 1);
24.     img_color = zeros(h,w,p);
25.     img_gray = zeros(h,w);
26.
27.     for H = 1:h
28.         for W = 1:w
29.             for P = 1:p
30.                 img_color(H,W,P) = round(round(img_org_data(H,W,P) / d) * d);
31.             end
32.             img_gray(H,W) = round(round(img_org_data(H,W) / d) * d);
33.         end
34.     end
35.
36.     subplot(4,3,j);
37.     imshow(uint8(img_color));
38.     num_str = L + "个灰度级（彩色图）";
39.     t = title(num_str);
40.     t.FontSize = 16;
41.
42.     subplot(4,3,6+j);
43.     imshow(uint8(img_gray));
44.     num_str = L + "个灰度级（灰度图）";
45.     t = title(num_str);
46.     t.FontSize = 16;
47. end
48.
49. %% Show the picture of Y,Cb,Cr
50. img_ycbcr = rgb2ycbcr(img_org_data);
51.
52. Y = img_ycbcr(:, :, 1);
53. Cb = img_ycbcr(:, :, 2);
54. Cr = img_ycbcr(:, :, 3);
55.
56. figure(3);
57. subplot(1,3,1);
58. imshow(Y);
59. title("Y");
60.
61. subplot(1,3,2);
62. imshow(Cb);
63. title("Cb");
64.
65. subplot(1,3,3);
66. imshow(Cr);
67. title("Cr");
68.
69. %% 下采样

```

```

70. figure(4);
71.
72. % 4:4:4
73. img_444(:,:,1) = Y;
74. img_444(:,:,2) = Cb;
75. img_444(:,:,3) = Cr;
76. img_444 = ycbcr2rgb(img_444);
77. subplot(2,2,1);
78. imshow(img_444);
79. title("4:4:4");
80.
81. % 4:2:2
82. Cb_422 = double(zeros(h,w));
83. Cr_422 = double(zeros(h,w));
84.
85.
86. for H = 1:1:h
87.     for W = 2:2:w
88.         Cb_422(H,W) = Cb(H,W);
89.         Cb_422(H,W-1) = Cb_422(H,W);
90.         Cr_422(H,W) = Cr(H,W);
91.         Cr_422(H,W-1) = Cr_422(H,W);
92.     end
93. end
94.
95. img_422(:,:,1) = Y;
96. img_422(:,:,2) = Cb_422;
97. img_422(:,:,3) = Cr_422;
98. img_422 = ycbcr2rgb(img_422);
99. subplot(2,2,2);
100. imshow(img_422);
101. title("4:2:2");
102.
103. % 4:1:1
104. Cb_411 = double(zeros(h,w));
105. Cr_411 = double(zeros(h,w));
106.
107. for H = 1:1:h
108.     for W = 4:4:w
109.         Cb_411(H,W) = Cb(H,W);
110.         Cb_411(H,W-1) = Cb_411(H,W);
111.         Cb_411(H,W-2) = Cb_411(H,W);
112.         Cb_411(H,W-3) = Cb_411(H,W);
113.         Cr_411(H,W) = Cr(H,W);
114.         Cr_411(H,W-1) = Cr_411(H,W);
115.         Cr_411(H,W-2) = Cr_411(H,W);
116.         Cr_411(H,W-3) = Cr_411(H,W);
117.     end
118. end
119. img_411(:,:,1) = Y;
120. img_411(:,:,2) = Cb_411;
121. img_411(:,:,3) = Cr_411;
122. img_411 = ycbcr2rgb(img_411);
123. subplot(2,2,3);
124. imshow(img_411);
125. title("4:1:1");
126.
127. % 4:2:0
128. Cb_420 = double(zeros(h,w));
129. Cr_420 = double(zeros(h,w));
130.
131. for H = 2:2:h

```

```

132.     for W = 2:2:W
133.         Cb_420(H,W) = Cb(H,W);
134.         Cb_420(H,W-1) = Cb_420(H,W);
135.         Cr_420(H,W) = Cr(H,W);
136.         Cr_420(H,W-1) = Cr_420(H,W);
137.     end
138.     Cb_420(H-1,:) = Cb_420(H,:);
139.     Cr_420(H-1,:) = Cr_420(H,:);
140. end
141.
142. img_420(:,:,1) = Y;
143. img_420(:,:,2) = Cb_420;
144. img_420(:,:,3) = Cr_420;
145. img_420 = ycbcr2rgb(img_420);
146. subplot(2,2,4);
147. imshow(img_420);
148. title("4:2:0");

```

5.2 DCT 变换

```

1. clear;
2. close all;
3.
4. %% Set Parameter
5. img_org = "img_org.jpg";
6. level = [2 4 8 64 128 256];
7.
8. %% 获取图片信息
9. img_data = imread(img_org);
10. % 旋转图片
11. for P = 1:3
12.     img_org_data(:,:,P) = img_data(:,:,P)';
13. end
14. [h,w,p] = size(img_org_data);
15.
16. figure(1);
17. imshow(img_org_data);
18.
19. %% 处理灰度级
20. figure(2);
21. for j = 1:6
22.     L = level(j);
23.     d = 255 / (L - 1);
24.     img_color = zeros(h,w,p);
25.     img_gray = zeros(h,w);
26.
27.     for H = 1:h
28.         for W = 1:w
29.             for P = 1:p
30.                 img_color(H,W,P) = round(round(img_org_data(H,W,P) / d) * d);
31.             end
32.             img_gray(H,W) = round(round(img_org_data(H,W) / d) * d);
33.         end
34.     end
35.
36.     subplot(4,3,j);
37.     imshow(uint8(img_color));
38.     num_str = L + "个灰度级（彩色图）";
39.     t = title(num_str);
40.     t.FontSize = 16;
41.

```

```

42.     subplot(4,3,6+j);
43.     imshow(uint8(img_gray));
44.     num_str = L + "个灰度级（灰度图）";
45.     t = title(num_str);
46.     t.FontSize = 16;
47. end
48.
49. %% Show the picture of Y,Cb,Cr
50. img_ycbcr = rgb2ycbcr(img_org_data);
51.
52. Y = img_ycbcr(:,:,1);
53. Cb = img_ycbcr(:,:,2);
54. Cr = img_ycbcr(:,:,3);
55.
56. figure(3);
57. subplot(1,3,1);
58. imshow(Y);
59. title("Y");
60.
61. subplot(1,3,2);
62. imshow(Cb);
63. title("Cb");
64.
65. subplot(1,3,3);
66. imshow(Cr);
67. title("Cr");
68.
69. %% 下采样
70. figure(4);
71.
72. % 4:4:4
73. img_444(:,:,1) = Y;
74. img_444(:,:,2) = Cb;
75. img_444(:,:,3) = Cr;
76. img_444 = ycbcr2rgb(img_444);
77. subplot(2,2,1);
78. imshow(img_444);
79. title("4:4:4");
80.
81. % 4:2:2
82. Cb_422 = double(zeros(h,w));
83. Cr_422 = double(zeros(h,w));
84.
85.
86. for H = 1:1:h
87.     for W = 2:2:w
88.         Cb_422(H,W) = Cb(H,W);
89.         Cb_422(H,W-1) = Cb_422(H,W);
90.         Cr_422(H,W) = Cr(H,W);
91.         Cr_422(H,W-1) = Cr_422(H,W);
92.     end
93. end
94.
95. img_422(:,:,1) = Y;
96. img_422(:,:,2) = Cb_422;
97. img_422(:,:,3) = Cr_422;
98. img_422 = ycbcr2rgb(img_422);
99. subplot(2,2,2);
100. imshow(img_422);
101. title("4:2:2");
102.
103. % 4:1:1

```

```

104. Cb_411 = double(zeros(h,w));
105. Cr_411 = double(zeros(h,w));
106.
107. for H = 1:1:h
108.     for W = 4:4:w
109.         Cb_411(H,W) = Cb(H,W);
110.         Cb_411(H,W-1) = Cb_411(H,W);
111.         Cb_411(H,W-2) = Cb_411(H,W);
112.         Cb_411(H,W-3) = Cb_411(H,W);
113.         Cr_411(H,W) = Cr(H,W);
114.         Cr_411(H,W-1) = Cr_411(H,W);
115.         Cr_411(H,W-2) = Cr_411(H,W);
116.         Cr_411(H,W-3) = Cr_411(H,W);
117.     end
118. end
119. img_411(:,:,1) = Y;
120. img_411(:,:,2) = Cb_411;
121. img_411(:,:,3) = Cr_411;
122. img_411 = ycbcr2rgb(img_411);
123. subplot(2,2,3);
124. imshow(img_411);
125. title("4:1:1");
126.
127. % 4:2:0
128. Cb_420 = double(zeros(h,w));
129. Cr_420 = double(zeros(h,w));
130.
131. for H = 2:2:h
132.     for W = 2:2:w
133.         Cb_420(H,W) = Cb(H,W);
134.         Cb_420(H,W-1) = Cb_420(H,W);
135.         Cr_420(H,W) = Cr(H,W);
136.         Cr_420(H,W-1) = Cr_420(H,W);
137.     end
138.     Cb_420(H-1,:) = Cb_420(H,:);
139.     Cr_420(H-1,:) = Cr_420(H,:);
140. end
141.
142. img_420(:,:,1) = Y;
143. img_420(:,:,2) = Cb_420;
144. img_420(:,:,3) = Cr_420;
145. img_420 = ycbcr2rgb(img_420);
146. subplot(2,2,4);
147. imshow(img_420);
148. title("4:2:0");

```