

第三章 经典数字图像水印算法

3.1 LSB 算法

LSB (Least Significant Bits) 算法，全称为最低有效位算法，是利用载体对象的二进制的最低一位（或几位）来进行秘密信息的隐藏。LSB 算法是隐写术中最常见的算法，多用于图像和声音载体之中^[9]。

(1) LSB 算法的可行性说明

- 图像/音频对象均可数字化为 $\{c_i\}$ 序列；
- 每一个对象 c_i 均由若干二进制位组成；
- 每一个对象 c_i 的二进制低位在某种意义上“不重要”。

对于具体图像来说：

- 去掉最低 1 位甚至最低 1-4 位对图像的整体视觉效果没有太大影响；
- 去掉最低 1 位对图像的统计特性影响不明显。

(2) LSB 算法嵌入基本步骤

- ① 将要隐藏的水印文本信息的转换为二进制数据；
- ② 读入载体图像，将图像矩阵的每个像素转换成二进制。用水印信息的二进制数据的每一比特位替换与之相对应的载体图像的最低有效位（LSB 位）；
- ③ 将载体图像的每一像素再由二进制转换为十进制，从而获得含有水印信息的图像。

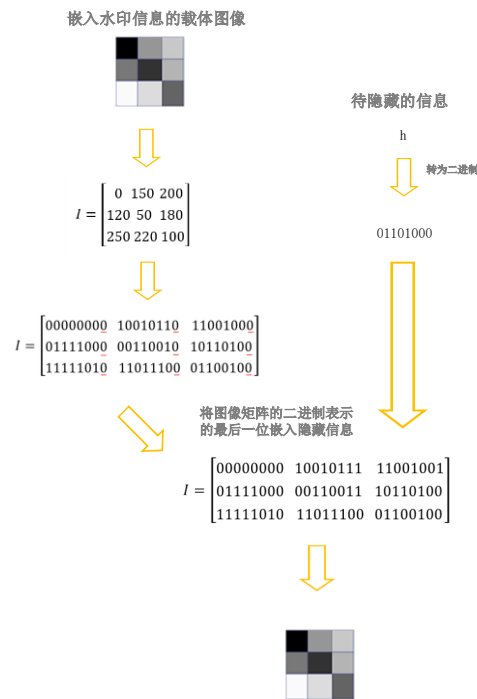


图 3.1 LSB 算法嵌入图解

(3) LSB 算法提取基本步骤

①将嵌入水印信息的载体图像的像素转换成二进制，依次取出每一个像素的二进制的最后一位，拼接成二进制数据。

③ 将得到的二进制数据转换成文本数据，便可得到隐藏的信息。

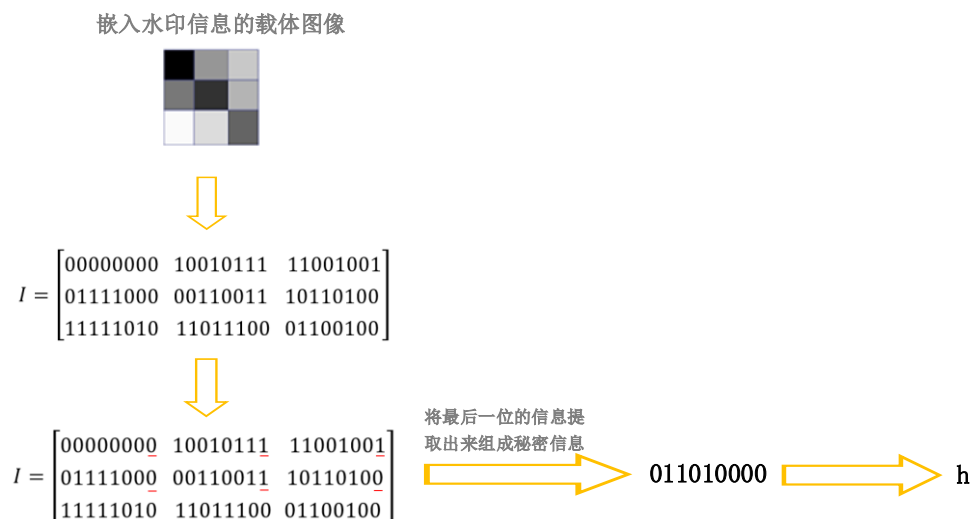


图 3.2 LSB 算法提取图解

3.2 DCT 算法

DCT 算法(Discrete Cosine Transform)，即离散余弦变换算法，它与离散傅里叶变换类似，但是只使用实数。DCT 算法于 1974 年被提出，被认为是对语音和图像信号进行变换的最佳方法。DCT 算法目前在图像编码中占有极其重要的地位，是 H.261、JPEG、MPEG、H.264 等国际公用的编码标准的重要基础。在视频压缩中，最常用的变换方法就是 DCT 算法。

在自然界中，大多数信号的能量都集中在余弦变换后的低频部分。并且人眼对于细节信息并不敏感，含有多数能量的低频部分几乎可以代表整幅图像，使用 DCT 算法进行信息隐藏就是在这个特点之上进行的。

(1) DCT 变换的原理

数字图像 S 是具有 M 行 N 列的一个矩阵。为了减弱或去除图像数据相关性，可以运用二维 DCT，将图像从空间域转换到 DCT 变换域。

①一维 DCT 变换

离散余弦变换

$$S(k) = D\{s\} = \frac{c(k)}{2} \sum_{j=0}^{N-1} s(j) \cos\left(\frac{(2j+1)k\pi}{2N}\right) \quad (3-1)$$

逆变换

$$S(k) = D^{-1}\{S\} = \sum_{j=0}^{N-1} \frac{C(j)}{2} S(j) \cos\left(\frac{(2j+1)k\pi}{2N}\right) \quad (3-2)$$

其中

$$C(j) = \begin{cases} \frac{1}{\sqrt{2}} & j = 0 \\ 1 & otherwise \end{cases} \quad (3-3)$$

②二维 DCT 变换

$$S(u, v) = \frac{2}{N} C(u) C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} s(x, y) \cos\left(\frac{\pi u(2x+1)}{2N}\right) \cos\left(\frac{\pi v(2y+1)}{2N}\right) \quad (3-4)$$

二维 DCT 逆变换

$$S(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u) C(v) S(u, v) \cos\left(\frac{\pi u(2x+1)}{2N}\right) \cos\left(\frac{\pi v(2y+1)}{2N}\right) \quad (3-5)$$

对于二维图像来说，s 表示空域(灰度)值，S 表示频率系数。

(2) DCT 变换的特点

在基于 DCT 的变换编码中，图像是先经分块（8×8 或 16×16）后再经过 DCT 变换，这种变换是局部的，只反映了图像某一部分的信息。当然也可以对整幅图像进行 DCT 变换，但是运算速度比分块 DCT 要慢。

图像经 DCT 后，得到的 DCT 图像有三个特点：

- ① 系数值全部集中在 0 值附近，动态范围很小；
- ② DCT 变换后图像能量集中在图像的低频部分，即 DCT 图像中不为零的系数大部分集中在一起；
- ③ 没有保留原图像块的精细结构，从中反映不了原图像块的边缘、轮廓等信息。

(3) DCT 信息嵌入算法

DCT 信息嵌入算法是以 DCT 变换为基础，对图像经过 DCT 变换后的 DCT 系数进行适当改变，以达到隐藏秘密信息的目的。将载体图像分成 8×8 的像素块，每一块只精确地编码（隐藏）一个秘密信息位。在每一块中，根据每一个图像块的两个 DCT 系数的相对大小来标明所隐藏的信息位。

生物学研究表明，人眼对图像平滑区域的变换比较敏感，而对纹理区域的变换不太敏感。经过离散余弦变换之后，图像信息集中在少数低频系数上，而纹理和边缘信息则在中高频系数中，所以低频系数的改变对图像视觉上的影响远大于高频系数。因此，在 DCT 信息嵌入的时候我们也是选择中频系数来隐藏信息，在 8×8 的像素块中，我们选择(4,1)和(3,2)这两个位置的中频系数来隐藏信息。当然，这只是其中一种选择，选择(4,2)和(2,4)等也是可以的。

嵌入过程开始时，首先随机地选择一个图像块 b_i ，它经二维 DCT 变换后得到 B_i ，用它对第 i 个消息比特 m_i 进行编码。发送者和接收者必须事先约定嵌入过程中使用的两个 DCT 系数的位置，为了达到隐藏的健壮性和不可察觉性，应该

在 DCT 的中频系数中选取，比如用 (u_1, v_1) 和 (u_2, v_2) 代表所选定的两个系数的坐标。嵌入过程为：

- 如果 $B_i(u_1, v_1) > B_i(u_2, v_2)$ ，就代表隐藏信息“1”；
- 如果 $B_i(u_1, v_1) < B_i(u_2, v_2)$ ，就代表“0”。

如果需要隐藏的信息位为“1”，但是 $B_i(u_1, v_1) < B_i(u_2, v_2)$ ，那么就把两个系数相互交换。最后发送者做二维逆 DCT 变换，将图像变回空间域，进行传输。

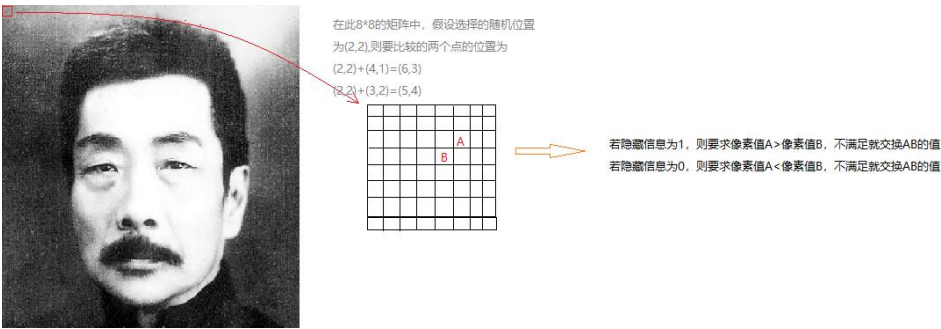


图 3.3 DCT 信息嵌入示意图

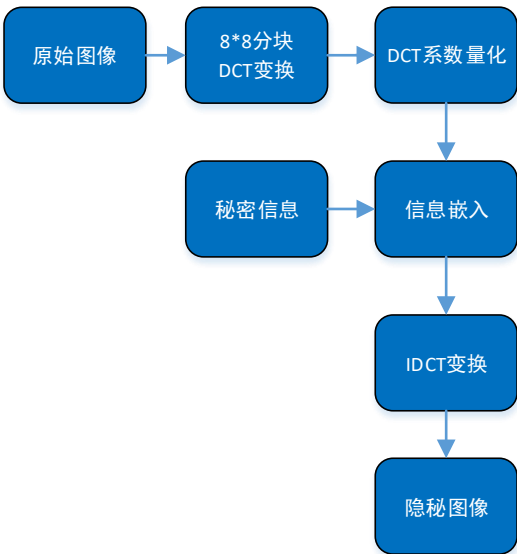


图 3.4 DCT 数字水印嵌入算法流程

(4) DCT 信息提取算法

DCT 信息提取是 DCT 信息嵌入的逆变换，将图像进行 DCT 变换后，从 DCT 系数中按照嵌入时的规则提取出隐藏信息。

- 如果 $B_i(u_1, v_1) > B_i(u_2, v_2)$ ，代表本信息块隐藏的信息为“1”；
- 如果 $B_i(u_1, v_1) < B_i(u_2, v_2)$ ，代表本信息块隐藏的信息为“0”。

将提取的信息拼接起来得到的就是隐藏的水印信息



图 3.5 DCT 数字水印提取算法流程

第四章 经典数字图像水印算法的改进

4.1 随机间隔算法

随机间隔算法是 LSB 算法的改进，发送者和接受者使用同一个密钥作为随机数生成器的种子，生成随机序列 $k_1, k_2, \dots, k_{l(m)}$ ，并进一步以此生成隐藏位置的指标集。

(1) 随机间隔法嵌入水印

随机间隔法嵌入水印的过程与 LSB 基本算法嵌入水印的过程基本相似，可以参见图 3.1，只是不再按顺序依次在载体图像的每一位像素中嵌入了，而是根据随机序列和步长来跳跃一段间隔，这也是为什么叫做随机间隔算法的原因。

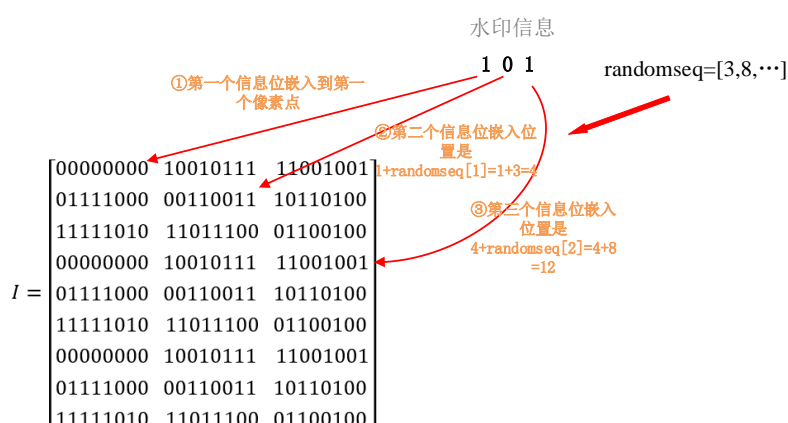


图 4.1 随机间隔法嵌入信息示意图

伪代码示意为：

使用种子 k 生成随机序列 k_i

$n \leftarrow k_l$

for $i=1 \dots, l(m)$ do

$s_i \leftarrow m_i$

$n \leftarrow n + k_{i+1}$

end for

(2) 随机间隔法提取水印

随机间隔算法提取水印是随机间隔法嵌入水印的逆操作，提取信息时使用与嵌入时相同的随机数序列，即可跳跃到相同的位置提取出隐藏信息。

伪代码示意为：

使用种子 k 生成随机序列 k_i

```

 $n \leftarrow k_l$ 

for  $i=1 \dots, l(m)$  do

     $m_i \leftarrow LSB(s_n)$ 

     $n \leftarrow n + k_{i+1}$ 

end for

```

(3) 随机间隔法性能分析

① 优点

- 随机确定水印信息比特嵌入的位置，提高了安全性，使水印位置不容易被定位。
- 因为有随机间隔步长的存在，水印信息的嵌入比基本 LSB 算法更为分散，水印嵌入对载体图片的统计特性影响更小，更不易被察觉。

② 缺陷

- 隐藏容量比基本 LSB 技术小。

(4) 随机间隔算法的步长对存储信息容量的影响

$$Capacity = \frac{Width * Height}{Step} \quad (4-1)$$

其中，Capacity 是存储信息的容量，也即存储信息的长度。Width 和 Height 是载体图片的长和宽，Width*Height 是载体图片的大小。Step 为随机间隔法的步长。

由此可见，在给定载体图片的情况下，随机间隔算法的步长与存储信息容量成反比，存储信息的容量也与随机间隔算法的步长成反比，二者双向影响，即

$$Step \propto \frac{1}{Capacity} \quad (4-2)$$

$$Capacity \propto \frac{1}{Step} \quad (4-3)$$

在系统设计中，在用户给定载体图片和隐藏信息后，我们可以计算出用户可以设定的最大步长，在不超过这个步长的情况下，都可以正常将信息隐写到载体图片中。

4.2 区域校验位算法

区域校验位算法也是一种 LSB 算法的改进。利用区域校验位算法隐藏信息，首先把载体分成几个不相交的区域，再利用校验位在每一个区域中隐藏 1 比特信息。

(1) 嵌入过程

根据设定的区域大小在载体图像上划分区域，根据要嵌入的载体信息 m_i ，计

算每一个区域的奇偶校验位 $p(I_i)$ 。

一个载体区域 I_i 的奇偶校验位定义为

$$p(I_i) = \sum_{j \in I_i} LSB(C_j) \bmod 2 \quad (4-4)$$

- 若 $p(I_i)$ 与 c 一致，则 $p(I_i)$ 已经隐藏了 m_i ；
- 若 $p(I_i)$ 与 m_i 不一致，则改变 I_i 中任意一个元素的 LSB，使得 $p(I_i) = m_i$ 。

(2) 提取过程


提取过程是嵌入过程的逆过程，根据嵌入时的区域大小在划分区域，依次在每块区域上计算其奇偶校验位 $p(I_i)$ ， $p(I_i)$ 就是本区域隐藏的信息。然后将每一块区域提取出的信息拼接在一起，即可得到隐藏的水印信息。

(3) 区域校验位算法示例

假设待隐藏信息为 TU，其 ASCII 码表示为

0101010001010101

截取载体图像的部分灰度值为



	1	2	3	4	5	6	7	8	9	10
1	60	76	84	127	115	77	52	37	18	13
2	82	110	108	135	123	79	48	31	13	13
3	109	131	168	154	139	65	53	19	23	12
4	157	163	184	161	137	63	52	19	23	11
5	192	188	198	167	121	49	42	16	26	16
6	150	150	159	123	97	29	28	10	29	25
7	163	169	174	130	90	21	22	8	31	30
8	164	169	169	118	107	35	31	13	34	30
9	194	197	203	158	129	54	45	22	39	32
10	189	194	214	180	142	64	53	28	44	36
11	210	220	204	220	187	129	84	60	32	17
12	220	213	210	182	160	108	71	53	30	20
13	226	197	218	134	117	76	51	42	27	25

图 4.2 截取载体图像的部分灰度值

选择每 2*2 个像素点为一个区域，利用其 LSB 校验位隐藏一位二进制信息。这里我们选取信息的前四位 0101，对应载体区域为

60	78		84	127		115	77		52	37
82	110		108	135		123	79		48	31

图 4.3 前 4 位信息对应的载体区域

将像素转换为二进制形式为

111100	1001110		1010100	1111111		1110011	1001101		110100	100101
1010010	1101110		1101100	10000111		1111011	1001111		110000	11111

图 4.4 前 4 位信息对应的载体区域的二进制表示

① 对于第一位信息位 0，计算第一块区域的奇偶校验位

$$I(1)=(0+0+0+0) \bmod 2 = 0$$

因为要嵌入的信息位也是 0，所以第一块区域的像素值不需要改变。

② 对于第二位信息位 1，计算第一块区域的奇偶校验位

$$I(2)=(0+1+0+1) \bmod 2 = 0$$

因为要嵌入的信息位是 1，所以在第二块区域的像素值中随机选一个将其二进制的最后一位设为 1，使得 $I(2)=(1+1+0+1) \bmod 2 = 1$ 。

以此类推，第三位信息位，第四位信息位按照同样的原则嵌入第三块和第四块区域。

嵌入信息后可能的结果为

60	78	84	126	115	77	52	37
82	110	108	135	123	79	49	31

图 4.5 嵌入信息后可能的结果

(4) 区域校验位算法性能分析

① 优点

- 载体区域中最多只需更改 1 个元素的 1 个比特位，这样秘密消息的嵌入对载体的统计特性改变最小。抵抗被动攻击的安全性较高。
- 与基本 LSB 算法相比，一个区域中只修改一个元素，水印嵌入对载体图片的统计特性影响更小，更不易被察觉。

② 缺陷

- 隐藏容量比基本 LSB 技术小。若区域大小为 N，则容量恰好缩小了 N 倍。
- 类似于 LSB 技术，该技术抵抗修改的能力依然很脆弱。

(5) 区域校验位算法的区域大小对存储信息容量的影响

$$ZoneSize = \frac{Width * Height}{Capacity} \quad (4-5)$$

其中，ZoneSize 为随机间隔法的步长，Width 和 Height 是载体图片的长和宽，Width*Height 是载体图片的大小，Capacity 是存储信息的容量，也即存储信息的长度。

由此看见，在给定载体图片的情况下，区域校验位算法的区域大小与存储信息容量成反比，存储信息的容量也与区域校验位算法的区域大小成反比，二者双向影响，即

$$ZoneSize \propto \frac{1}{Capacity} \quad (4-6)$$

$$Capacity \propto \frac{1}{ZoneSize} \quad (4-7)$$

在系统设计中，在用户给定载体图片和隐藏信息后，我们可以计算出用户可

以设定的最大区域大小，在不超过这个区域大小的情况下，都可以正常将信息隐写到载体图片中。

4.3 图像降级算法及其改进

4.3.1 图像降级算法

图像降级算法是 LSB 算法的改进，但改进的不是算法性能而是载体种类。图像降级算法要做的是将一个水印图片嵌入到另一张图片里去。在图像中隐藏图像和在图像中隐藏文本有很大的不同，因为图像文件和文本文件相比要大得多，图像水印也难以直接使用 LSB 算法进行嵌入。

(1) 图像降级算法的原理

给定一个同样尺寸的载体图像 c 和秘密图像 m ，发送者将载体图像灰度值的四个最低位比特位替换成秘密图像的四个最高比特位。接收者从隐藏后的图像中提取四个最低的比特位，从而获得秘密图像的最高四个比特位，将四个低比特位填充为 0。

重构后的图像实质上只有四个比特位，是秘密图像的一种近似。视觉上，重构的图像和原始图像非常相似。

图像降级算法可看作是 LSB 算法的一种推广，但是隐藏位由 1 位变成了 4 位，同时对水印图片也有要求，即水印图片的大小要小于等于嵌入图片。

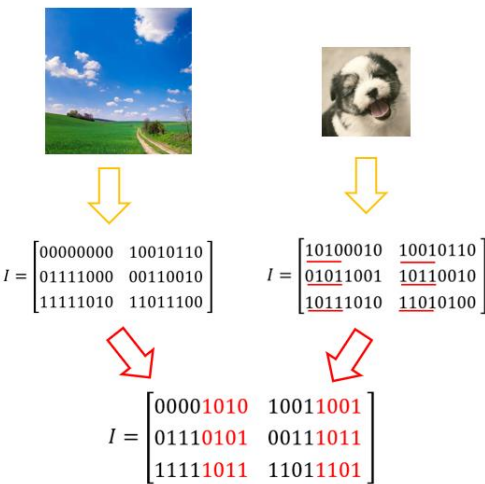


图 4.6 图像降级算法原理图

(2) 选择替换位数为 4 的原因

用 n 表示载体图像用来隐藏信息的低位数，研究人员将 n 的不同值都做了实验，从实验结果来看，选择 4 位能够最大程度地保证对嵌入水印的载体图像和提取出的水印图像的图片质量影响最小。

当 n 较小时，提取出来的水印图像的质量会受到较大影响。



图 4.7 $n=3$ 时的图像质量对比

当 n 较大时，嵌入水印后的伪装图像的质量有较大影响。



图 4.8 $n=5$ 时的图像质量对比

(3) 图像降级算法的缺陷

由于图像降级算法是将载体图片的后四个比特位替换掉了，相当于是替换了载体图片的 $1/2$ 位，所以有些时候嵌入水印后的图片可能会产生一些比较轻微的肉眼可见的变化。

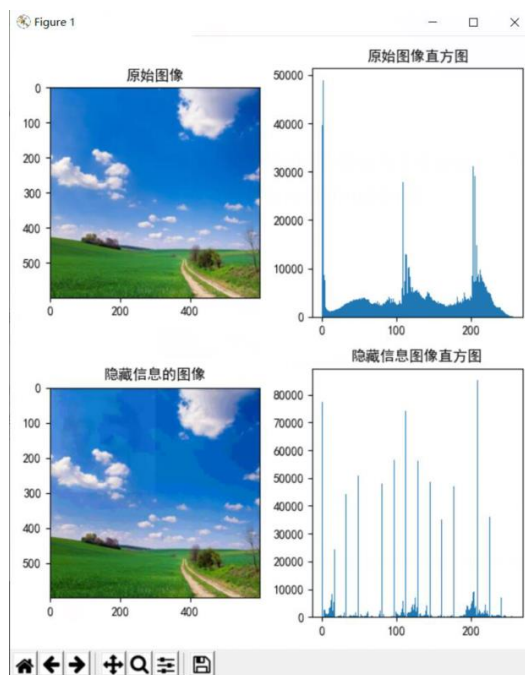


图 4.9 图像降级算法产生的轻微肉眼可见的变化

为了减少对载体图片质量的影响，我自己思考提出了一种对图像降级算法的改进方法。

4.3.2 图像降级算法的改进

要想减少对载体图像的影响，就要减少每个像素嵌入的位数，因此我们可以将八位的二进制数分成四块，每块分别加入到载体图片上去，这样能够保证当载体图片的长宽大于等于水印图片的长宽的两倍的时候，水印图片的信息不会丢失。

因为向图片中嵌入图片水印，嵌入的图片的比例不能大于隐藏信息的图片。又因为采取的算法是将八位的二进制数分成四块，所以嵌入的图片的大小要小于等于隐藏信息图片的 $1/4$ 。

所以，图像降级算法的改进是在提升了性能的基础上，减少了可以存储的信息容量。

(1) 水印嵌入的步骤

- ①遍历水印图片的各个像素点，将每个像素点转为 8 位二进制的形式；
- ②通过位运算的操作将 8 位的二进制分成四块，每块两位，嵌入到载体图片的 4 个像素中。

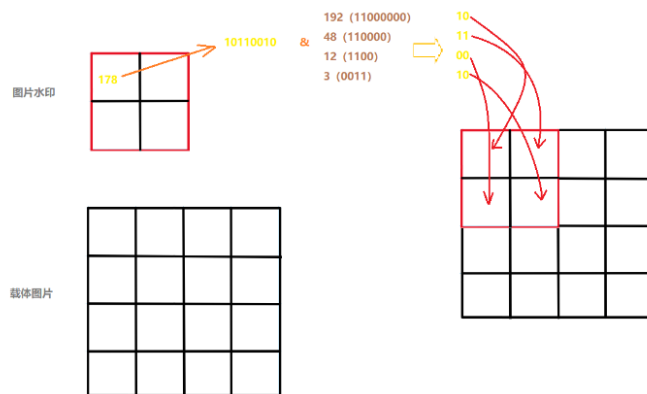


图 4.10 图像降级算法的改进演示图

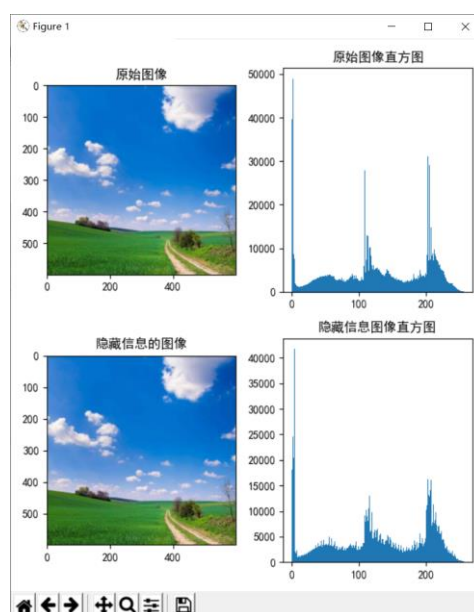


图 4.11 改进的图像降级算法对载体图像的影响很小

(2) 水印提取步骤

- ①每四个像素一组遍历嵌入水印的图片，提取水印信息；
- ②将每四个像素点提取的信息拼接成完整的信息。

第五章 可视化数字水印系统的实现

5.1 界面布局

基于数字图像的可视化水印系统按照水印算法的不同主要分为空间域水印和变换域水印两大类。空间域水印以 LSB 算法——最低有效位算法为代表，变换域水印以 DCT 算法——离散余弦变换算法为代表^[10]。

5.1.1 空间域水印

空间域水印分为三个部分：LSB 基本算法、LSB 算法改进和图像降级算法及其改进。其中 LSB 图片水印和 LSB 算法改进均是基于 LSB 基本算法所进行的推广和改进。

(1) LSB 基本算法

LSB 基本算法就是基于经典 LSB 算法理论进行文本水印嵌入和提取。LSB 基本算法包括 LSB 基本算法水印嵌入和 LSB 基本算法水印提取，可以实现将信息隐藏在图片中和从隐藏信息的图片中提取信息的功能。

(2) LSB 算法改进

LSB 算法改进包括 LSB 随机间隔算法和 LSB 区域校验位算法，二者均能够减小水印嵌入对载体图片统计特性的影响。

① LSB 随机间隔算法

LSB 随机间隔算法包括随机间隔水印法嵌入和随机间隔法水印提取。LSB 随机间隔法算法水印嵌入由用户选择图片和隐藏信息，对图像进行随机间隔的 LSB 隐写后，将秘密信息写入，然后绘制原始图像和隐写后的图像的直方图对比，并保存隐写后的图像。随机间隔算法步长由用户输入。

LSB 随机间隔法水印提取由用户选择要从中提取信息的图片和提取信息的保存路径，读取图像并提取出信息并保存到用户选择的路径。

② LSB 区域校验位算法

LSB 区域校验位算法包括区域校验位算法水印嵌入和区域校验位算法水印提取。

区域校验位算法水印嵌入由用户选择图片和隐藏信息，对图像进行区域校验位的 LSB 隐写后，将秘密信息写入，然后绘制原始图像和隐写后的图像的直方图对比，并保存隐写后的图像。区域校验位算法的区域大小由用户输入。

区域校验位算法水印提取由用户选择要从中提取信息的图片和提取信息的保存路径，读取图像并提取出信息并保存到用户选择的路径。

(3) 图像降级算法及其改进

① 图像降级算法

图像降级算法包括图像降级算法水印嵌入和图像降级算法水印提取，可以实

现将图片水印嵌入图片当中的功能。图像降级算法水印嵌入由用户选择载体图片和水印图片，对载体图像进行图像降级隐写，之后绘制原始图像和隐写后的图像的直方图对比，并保存隐写后的图像。图像降级水印提取由用户选择要从中提取信息的图片和提取信息的保存路径，读取图片并提取出信息并保存到用户选择的路径当中。

由于图片的信息量一般要高于文本的信息量，所以图像降级的嵌入和提取耗时要长一些。

② 图像降级算法改进

图像降级算法改进的界面布局和操作与图像降级算法完全相同，只是改进了对载体图像的画质的损伤，得到的嵌入水印的图像的效果要优于图像降级算法。

5.1.2 变换域水印

变换域水印包括 DCT 水印嵌入和 DCT 水印提取。

DCT 水印嵌入由用户选择图片和隐藏信息，对图像进行离散余弦变换后，将秘密信息写入，然后绘制原始图像和隐写后的图像的直方图对比，并保存隐写后的图像。

DCT 提取由用户选择提取信息的保存路径，程序将读取 DCT 隐写时保存的图像并提取出信息并保存到用户选择的路径。

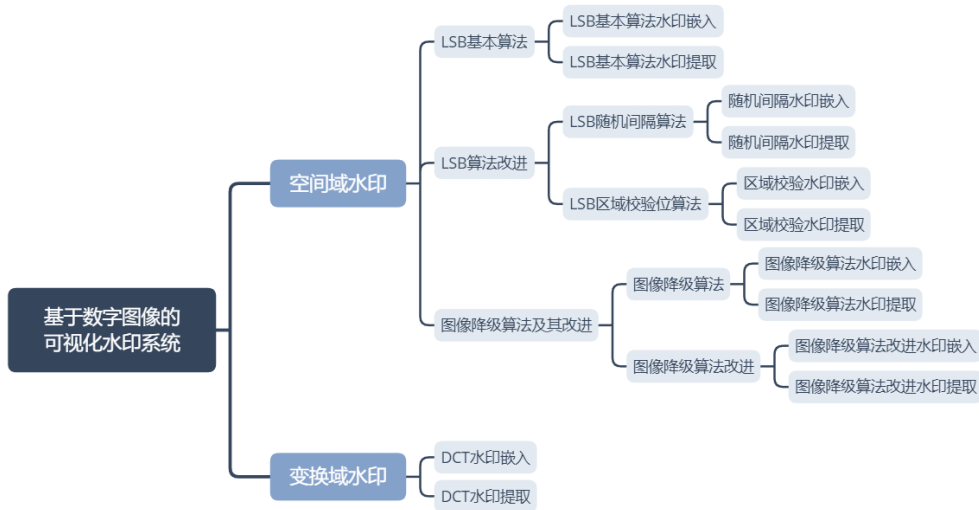


图 5.1 可视化数字水印系统分层模块图

5.2 程序运行界面

5.2.1主界面

主界面包括“空间域水印”和“变换域水印”按钮，在按钮下方是关于空间

域水印和变换域水印的组成的介绍。

空间域水印包含 LSB 水印嵌入和提取、LSB 算法改进、图像降级算法及其改进等功能。变换域水印包含 DCT 隐写和 DCT 提取。



图 5.2 可视化数字水印系统主界面

5.2.2 空间域水印界面

空间域水印界面包括 LSB 基本算法、LSB 算法改进和图像降级算法及其改进三个部分，点击按钮可以进行相应的功能，右侧分别是它们的介绍。

LSB 基本算法包括 LSB 基本算法水印嵌入和 LSB 基本算法水印提取，可以实现将信息隐藏在图片中和从隐藏信息的图片中提取信息的功能。

LSB 算法改进包括随机间隔法和区域校验位算法，在 LSB 算法的基础上，减小了水印嵌入对载体图片统计特性的影响。

图像降级算法及其改进包括图像降级算法和图像降级算法改进，可以实现将图片水印嵌入到图片当中的功能。



图 5.3 可视化数字水印系统空间域水印界面

（1）LSB 基本算法

LSB 基本算法包括 LSB 基本算法水印嵌入和 LSB 基本算法水印提取两个部分。点击按钮可以进行相应的功能，右侧分别是它们的介绍。

LSB 基本算法水印嵌入由用户选择图片和隐藏信息，对图像进行最低有效位隐写后，将秘密信息写入，然后绘制原始图像和隐写后的图像的直方图对比，并保存隐写后的图像。

LSB 基本算法水印提取由用户选择提取信息的保存路径，程序将读取 LSB 隐写时保存的图像并提取出信息并保存到用户选择的路径。

在进行 LSB 基本算法水印提取之前，需要先在下方输入提取信息的长度。



图 5.4 可视化数字水印系统 LSB 基本算法界面

（2）LSB 算法改进

LSB 算法改进包括“LSB 随机间隔法”和“LSB 区域校验位算法”两个部分。点击按钮可以进行相应的功能，右侧分别是它们的介绍。

LSB 随机间隔法包括随机间隔法水印嵌入和随机间隔法水印提取。

LSB 区域校验位算法包括区域校验位算法水印嵌入和区域校验位算法水印提取。



图 5.5 可视化数字水印系统 LSB 算法改进界面

①LSB 随机间隔法

LSB 随机间隔法包括“LSB 随机间隔法水印嵌入”和“LSB 随机间隔法水印提取”两个功能。点击按钮可以进行相应的功能，下方分别是它们的介绍。

随机间隔法水印嵌入由用户选择图片和隐藏信息，对图像进行随机间隔的 LSB 隐写后，将秘密信息写入，然后绘制原始图像和隐写后的图像的直方图对比，并保存隐写后的图像。随机间隔的步长在上方由用户输入。

随机间隔法水印提取由用户选择提取信息的保存路径，程序将使用同样的随机种子，读取随机间隔法水印嵌入时保存的图像并提取出信息并保存到用户选择的路径。

在用户进行随机间隔法嵌入和提取之前，需要首先在上方设置随机间隔的步长。如果用户不设置，步长默认为 2。如果步长设置得过长，以至于信息无法完整在图片中隐藏，程序会弹出消息框提示用户步长设置过长。

在用户进行随机间隔法提取之前，还需要在下方输入提取信息的长度。



图 5.6 可视化数字水印系统 LSB 随机间隔法界面

②LSB 区域校验位算法

LSB 区域校验位算法包括“LSB 区域校验位算法水印嵌入”和“LSB 区域校验位算法水印提取”。点击按钮可以进行相应的功能，下方是它们的介绍。

区域校验位算法水印嵌入由用户选择图片和隐藏信息，对图像进行区域校验位的 LSB 隐写后，将秘密信息写入，然后绘制原始图像和隐写后的图像的直方图对比，并保存隐写后的图像。区域校验位算法的区域大小由用户输入。

区域校验位算法水印提取由用户选择提取信息的保存路径，读取区域校验位算法水印嵌入时保存的图像并提取出信息并保存到用户选择的路径。

在用户进行区域校验位算法嵌入和提取之前，需要首先在上方设置区域校验位的区域大小。如果用户不设置，区域大小默认为 4。如果区域大小设置得过长，以至于信息无法完整在图片中隐藏，程序会弹出消息框提示用户区域大小设置过长。

在用户进行区域校验位算法提取之前，还需要在下方输入提取信息的长度。



图 5.7 可视化数字水印系统 LSB 区域校验位算法界面

(3) 图像降级算法及其改进

图像降级算法及其改进包括“图像降级算法”和“图像降级算法改进”两个部分。点击按钮可以进行相应的功能，右侧分别是它们的介绍。

图像降级算法包括图像降级算法水印嵌入和图像降级算法水印提取。

图像降级算法改进包括图像降级算法改进水印嵌入和图像降级算法改进水印提取。



图 5.8 可视化数字水印系统图像降级算法及其改进界面

① 图像降级算法

图像降级算法包括“图像降级算法水印嵌入”和“图像降级算法水印提取”两个功能。点击按钮可以进行相应的功能，右侧分别是它们的介绍。

图像降级算法水印嵌入由用户选择载体图片和水印图片，将载体图片的四个最低为比特位替换成水印图片的四个最高比特位，然后绘制原始图像和隐写后的图像的直方图对比，并保存隐写后的图像。

图像降级算法水印嵌入水印提取由用户选择要提取信息的图片和提取信息的保存位置，程序读取要提取信息的图片，提取出隐藏的图片并保存。



图 5.9 可视化数字水印系统图像降级算法界面

②图像降级算法改进

图像降级算法改进包括“图像降级算法改进水印嵌入”和“图像降级算法改进水印提取”两个功能。点击按钮可以进行相应的功能，右侧分别是它们的介绍。

图像降级算法改进水印嵌入由用户选择载体图片和水印图片，将水印图片的信息的八位的二进制数分成四块，每块分别加入到载体图片上去，然后绘制原始图像和隐写后的图像的直方图对比，并保存隐写后的图像。

图像降级算法水印嵌入水印提取由用户选择要提取信息的图片和提取信息的保存位置，程序读取要提取信息的图片，提取出隐藏的图片并保存。



图 5.10 可视化数字水印系统图像降级算法改进界面

5.2.3 变换域水印界面

变换域水印包括“DCT 水印嵌入”和“DCT 水印提取”两个功能。点击按钮可以进行相应的功能，下方是它们的介绍。

DCT 水印嵌入由用户选择图片和隐藏信息，对图像进行离散余弦变换后，将秘密信息写入，然后绘制原始图像和隐写后的图像的直方图对比，并保存隐写

后的图像。

DCT 提取由用户选择提取信息的保存路径，程序将读取 DCT 隐写时保存的图像并提取出信息并保存到用户选择的路径。



图 5.11 可视化数字水印系统变换域水印界面

5.3 系统功能展示

5.3.1 空间域水印

(一) LSB 基本算法

(1) LSB 基本算法水印嵌入

首先点击“LSB 基本算法水印嵌入”按钮，进行嵌入操作。



图 5.12 LSB 基本算法界面

下面会弹出提示,请用户选择要进行 LSB 隐写的图像。

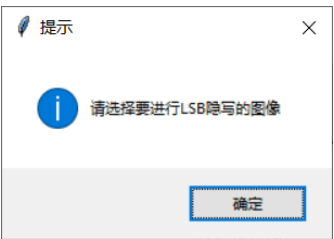


图 5.13 提示用户选择 LSB 隐写图像界面

点击“确定”，进行选择要隐写的图像，点击“打开”。

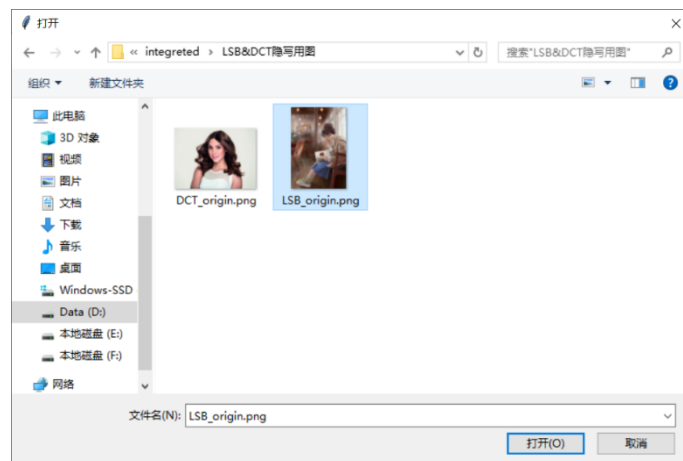


图 5.14 选择 LSB 隐写图像界面



图 5.15 载体图像

下面会提示用户选择要隐藏的信息，用户需要提前将要隐藏的信息放入一个 txt 文本文件中，然后选择这个文件，点击打开。

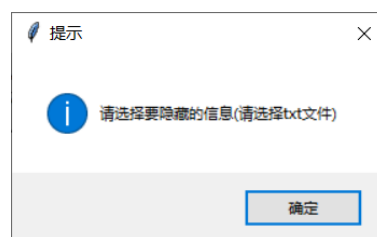


图 5.16 提示用户选择要隐藏的信息界面

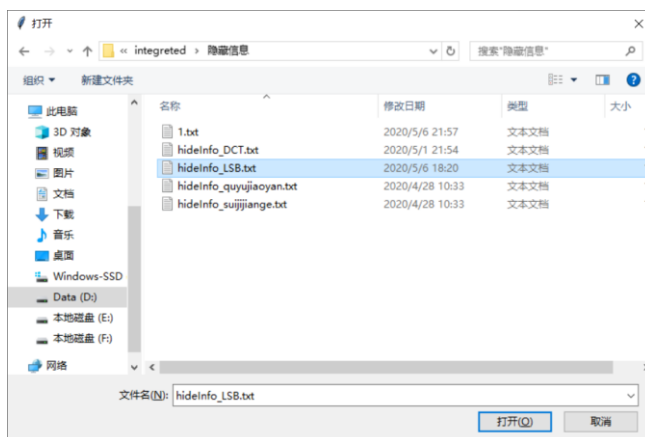


图 5.17 选择隐藏信息的界面

这是我们准备隐写入图像中的信息，

“Visual watermarking system based on digital image by hxx”



图 5.18 待隐写的秘密信息

下面程序就会将隐藏信息用 LSB 算法嵌入到用户选择的图片中。隐写成功后会弹出提示，告诉用户隐写已完成，并且将隐写后的图像保存在程序所在的目录下，命名为用户选择文件的文件名+_LSB-generate。

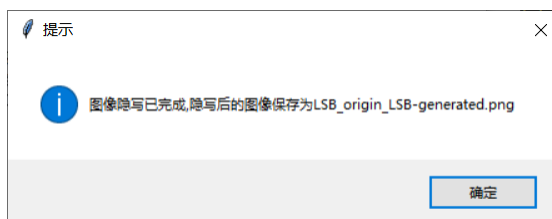


图 5.19 提示用户图像隐写已完成

点击确定后，会出现原始图像和隐藏信息的图像的对比图，并绘制它们的直方图。

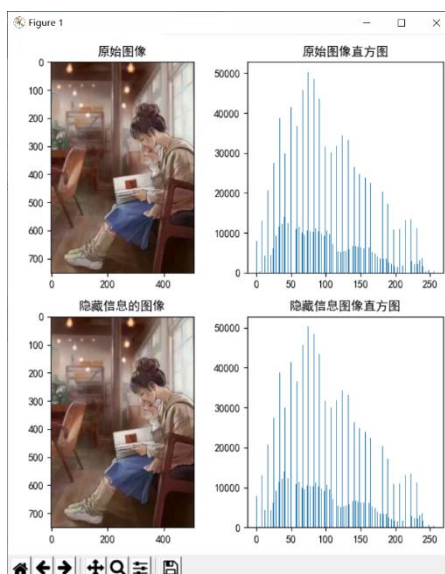


图 5.20 原始图像和隐藏信息的图像对比图以及直方图

(2) LSB 基本算法水印提取

首先预计要提取的信息的长度，输入提取信息的长度。注意这里的提取信息的长度是信息转换为二进制后的比特长度。

以我们嵌入的信息为例，“Visual watermarking system based on digital image by hxx”这段文本，转换为二进制后是

“01010110011010010111001101110101011000010110110000100000011101110110
0001011101000110010101110010011011010110000101110010011010110110100101
1011100110011100100000011100110111100101110011011101000110010101101101
0010000001100010011000010111001101100101011001000010000001101111011011
1000100000011001000110100101100111011010010111010001100001011011000010
0000011010010110110101100001011001110110010100100000011000100111100100
100000011010000111100001111000”

其信息长度为 448，在这里我们输入 448，点击“输入提取信息的长度”按钮。



图 5.21 输入提取信息的长度

会弹出提示，提取信息长度已经设置成功。

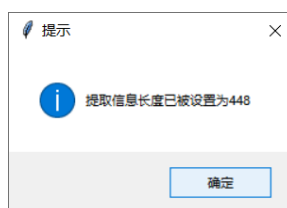


图 5.22 提取信息长度已设置的提示框

之后点击“LSB 基本算法水印提取”按钮。



图 5.23 点击“LSB 基本算法水印提取”按钮

按照弹出的提示，选择要提取隐藏信息的图像，选择后点击打开。

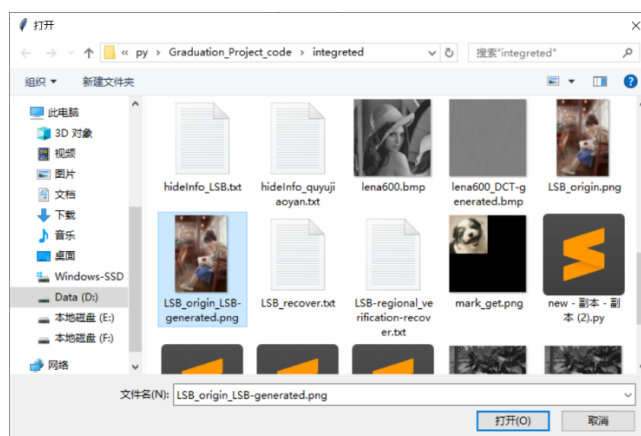


图 5.24 选择提取隐藏信息的图像

选择要将提取的信息保存的位置，点击确定。

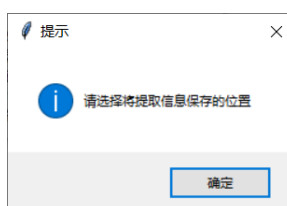


图 5.25 提示用户选择隐藏信息的保存位置

选择好保存的文件夹后点击“选择文件夹”。

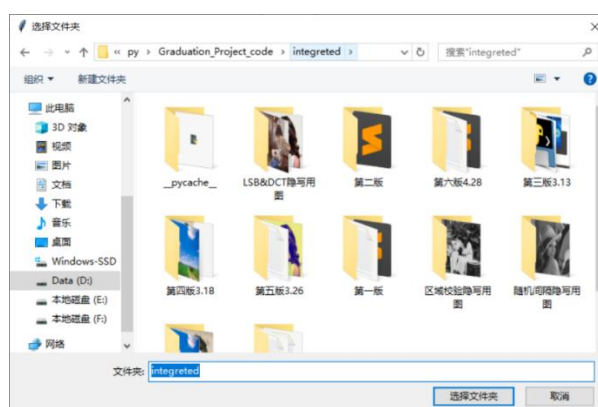


图 5.26 选择隐藏信息的保存位置

之后弹出提示，隐藏信息已提取，保存为 LSB_recover.txt。

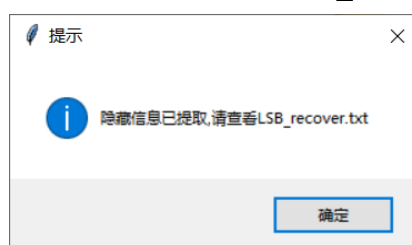


图 5.27 隐藏信息已提取提示框

来到选择保存信息的文件夹，打开 LSB_recover.txt，可以看到信息已被正确提取。

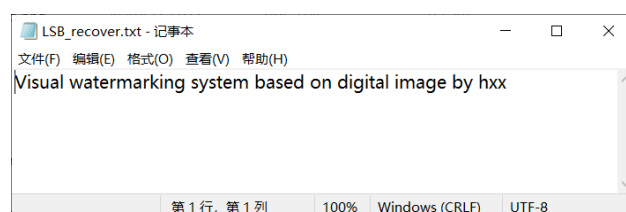


图 5.28 提取的隐藏信息

当然，用户在提取信息的时候，可能并不知道信息的具体长度，这就需要用用户根据提取出来的内容进行尝试几次，就可以正确提取，例如，以上面的信息提取为例，当我们输入提取信息长度为 100 时，提取的信息为



图 5.29 提取信息的长度输入较小时的结果

可以看到此时有效的信息似乎还没有结束。
当输入提取信息长度为 600 时，提取的信息为



图 5.30 提取信息的长度输入较大时的结果

可以看到有效的信息到“hxx”就结束了，后面就是乱码了，所以我们可以确定隐藏的信息就是“Visual watermarking system based on digital image by hxx”。

（二）LSB 算法改进

（1）LSB 随机间隔法

①LSB 随机间隔法水印嵌入

首先设置随机间隔的步长（如果用户没有设置，那么默认为 2），我们假定设置为 5。



图 5.31 设置随机间隔的步长

程序会弹出提示，随机间隔步长已被设置为 5。

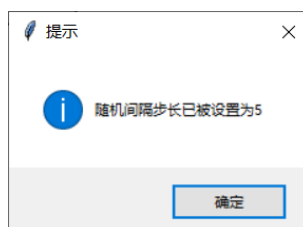


图 5.32 提示随机间隔步长已被设置

当然，随机间隔的步长不能无限大，必须要能保证隐藏信息能够被全部嵌入

到载体图片中。如果步长设置得过长，程序会弹出提示，不过要等用户给定载体图片和隐藏信息后才能判断出来。

点击“LSB 随机间隔法水印嵌入”按钮，进行嵌入操作。



图 5.33 点击“LSB 随机间隔法水印嵌入”按钮

下面会弹出提示,请用户选择要进行 LSB 随机间隔隐写的图像。点击“确定”进行选择要隐写的图像。选择图像后点击“打开”。



图 5.34 LSB 随机间隔法的载体图像

下面会提示用户选择要隐藏的信息，用户需要提前将要隐藏的信息放入一个 txt 文本文件中，然后选择这个文件。

这是我们准备隐写入图像中的信息，

“Visual watermarking system based on digital image by hxx” 选择好信息文件后，点击“打开”。

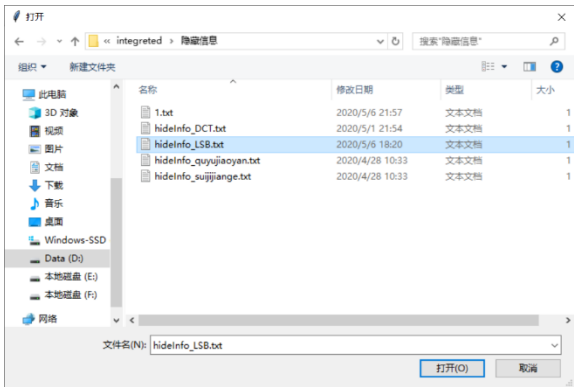


图 5.35 打开水印信息

此时，如果给定的步长设置不合适，程序就会弹出提示，告诉用户步长设置过大，请重新设置。



图 5.36 提示步长设置过大

下面程序就会将隐藏信息用随机间隔算法嵌入到用户选择的图片中。隐写成功后会弹出提示，告诉用户隐写已完成，并且将隐写后的图像保存在程序所在的目录下，命名为用户选择文件的文件名+_random_interval-generated。

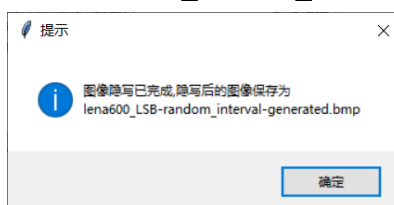


图 5.37 提示图像隐写完成

点击确定后，会出现原始图像和隐藏信息的图像的对比图，并绘制它们的直方图。

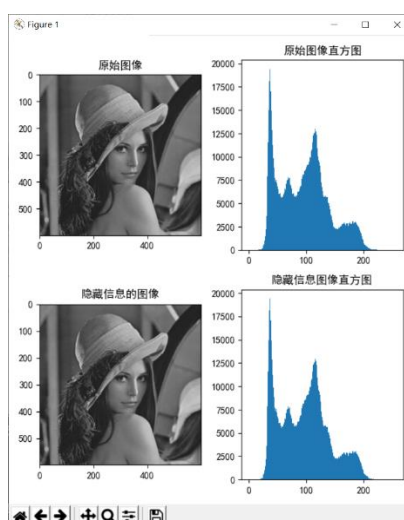


图 5.38 原始图像和隐藏信息图像的对比图以及直方图

②LSB 随机间隔法水印提取

首先也要设置随机间隔的步长，并且，要想正确提取信息，提取时设置的步长要和隐写时设置的步长相同。

然后在下方输入提取信息的长度。在这里我们输入 448，点击“输入提取信息的长度”按钮。之后会弹出提示，提取信息长度已经设置成功。

之后点击“LSB 随机间隔法水印提取”按钮。



图 5.39 点击“LSB 随机间隔法水印提取”按钮

按照弹出的提示，选择要提取隐藏信息的图像和选择要将提取的信息保存的位置。

之后弹出提示，隐藏信息已提取，保存为 LSB-random_interval-recover.txt 来到选择保存信息的文件夹，打开 LSB-random_interval-recover.txt，可以看到，信息已被正确提取。

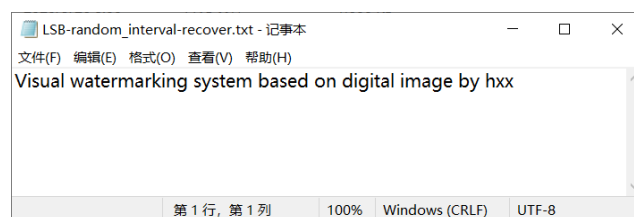


图 5.40 提取的隐藏信息

(2) LSB 区域校验位算法

①LSB 区域校验位算法水印嵌入

首先设置区域校验位算法的区域大小（如果用户没有设置，那么默认为 4），我们假定设置为 5。

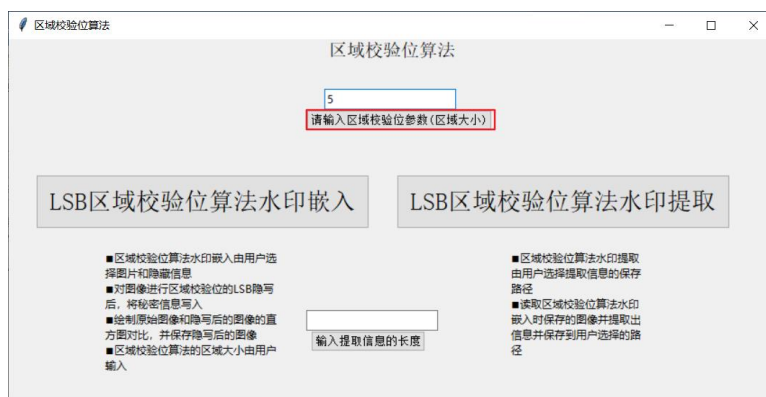


图 5.41 设置区域校验位的区域大小

程序会弹出提示，区域大小已被设置为 5。

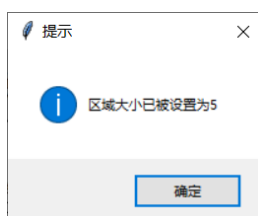


图 5.42 提示区域大小已被设置

当然，区域大小不能无限大，必须要能保证隐藏信息能够被全部嵌入到载体图片中。如果区域大小设置得过大，程序会弹出提示，不过要等用户给定载体图片和隐藏信息后才能判断出来。

点击“LSB 随机间隔法水印嵌入”按钮，进行嵌入操作。



图 5.43 点击“LSB 区域校验位算法水印嵌入”按钮

下面会弹出提示,请用户选择要进行 LSB 区域校验位隐写的图像。



图 5.44 LSB 区域校验位算法载体图像

下面会提示用户选择要隐藏的信息。用户需要提前将要隐藏的信息放入一个 txt 文本文件中，然后选择这个文件。

这是我们准备隐写入图像中的信息

“Visual watermarking system based on digital image by hxx”，选择好信息文件后，点击“打开”。

此时，如果给定的区域大小设置不合适，程序就会弹出提示，告诉用户区域大小设置过大，请重新设置。

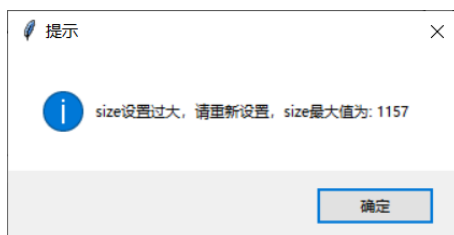


图 5.45 提示区域设置过大

下面程序就会将隐藏信息用区域校验位算法，嵌入到用户选择的图片中。隐写成功后会弹出提示，告诉用户隐写已完成，并且将隐写后的图像保存在程序所在的目录下，命名为用户选择文件的文件名+_LSB-regional_verification-generated。

点击确定后，会出现原始图像和隐藏信息的图像的对比图，并绘制它们的直方图。

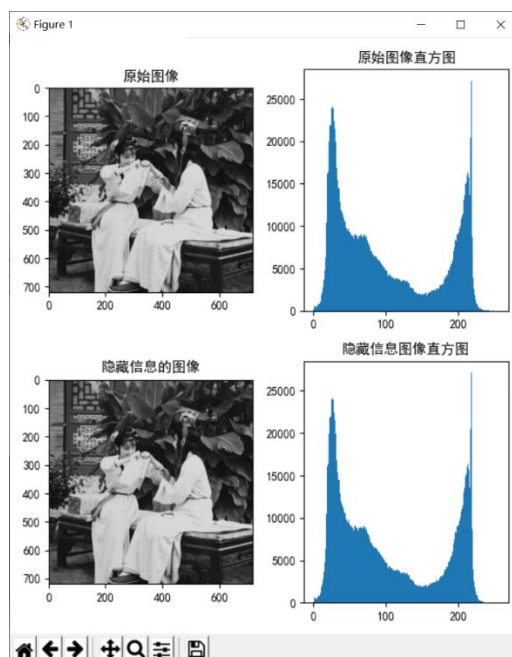


图 5.46 原始图像和隐藏信息图像的对比图以及直方图

②LSB 区域校验位算法水印提取

首先也要设置区域校验位算法的区域大小，并且，要想正确提取信息，提取时设置的区域大小要和隐写时设置的区域大小相同。

然后在下方输入提取信息的长度。在这里我们输入 448，点击“输入提取信息的长度”按钮。

之后点击“LSB 区域校验位算法水印提取”按钮。



图 5.47 点击“LSB 区域校验位算法水印提取”按钮

按照弹出的提示，选择要提取隐藏信息的图像和要将提取的信息保存的位置，之后弹出提示，隐藏信息已提取，保存为 LSB-regional_verification-recover.txt。来到选择保存信息的文件夹，打开 LSB-regional_verification-recover.txt，可以看到

信息已被正确提取。



图 5.48 提取的隐藏信息

(三) 图像降级算法及其改进

(1) 图像降级算法

①图像降级算法水印嵌入

首先点击“图像降级算法水印嵌入”按钮，进行嵌入操作。



图 5.49 点击“图像降级算法水印嵌入”按钮

下面会弹出提示,请用户选择被嵌入水印的载体图像。向图片中添加图片水印，对嵌入图片的大小有限制，嵌入的图片的大小不能大于隐藏信息的图片。

这是我们准备的载体图片和嵌入载体图片的水印图片。



图 5.50 载体图片

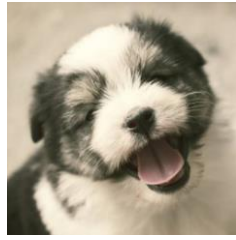


图 5.51 水印图片

选择载体图片后点击“打开”。之后会提示用户选择要隐写的图片，选择好隐写图片后，点击“打开”。

下面程序就会将要隐写的图片，用 LSB 算法，嵌入到用户选择的图片中。隐写成功后会弹出提示，告诉用户隐写已完成，并且将隐写后的图像保存在程序所在的目录下，命名为用户选择文件的文件名+_image_with_mark。

点击确定后，会出现原始图像和隐藏信息的图像的对比图，并绘制它们的直方图。

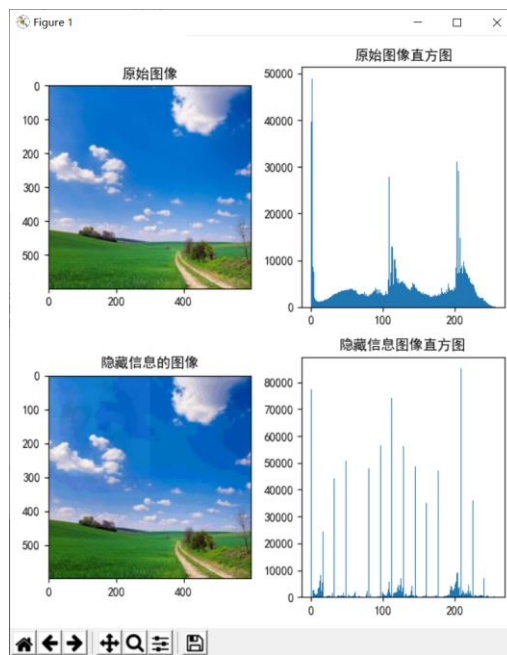


图 5.52 原始图像和隐藏信息的图像对比图以及直方图

③ 图像降级算法水印提取

点击“图像降级算法水印提取”按钮。

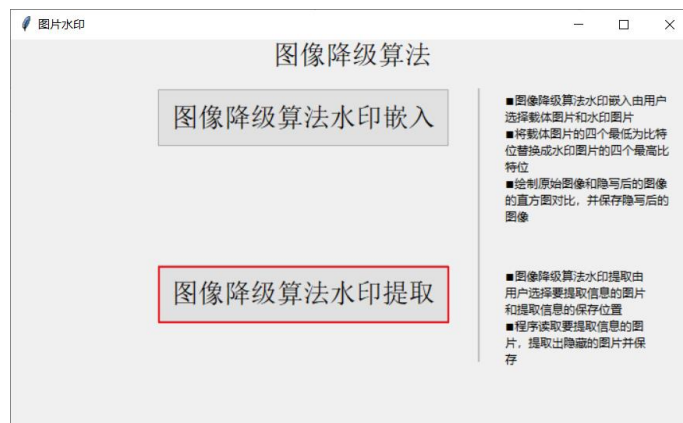


图 5.53 点击“图像降级算法水印提取”按钮

按照弹出的提示，选择要提取隐藏信息的图像和要将提取的信息保存的位置。之后弹出提示，水印图片已提取，保存为 mark_get.png。

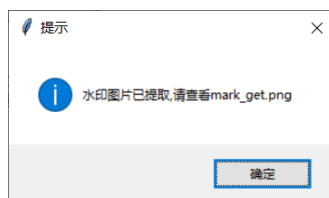


图 5.54 提示水印图片已提取

来到选择保存信息的文件夹，打开 mark_get.png，可以看到隐藏的图片已被正确提取。

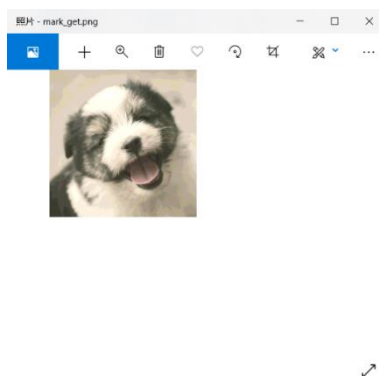


图 5.55 提示的水印图片

(2) 图像降级算法改进

图像降级算法改进的水印嵌入和提取的操作过程与图像降级算法的水印嵌入和提取完全相同，最终的区别在于得到的隐写信息的图像的画质会有所提升，

而水印图像的长和宽需要分别小于等于载体图片长和宽的 1/4。

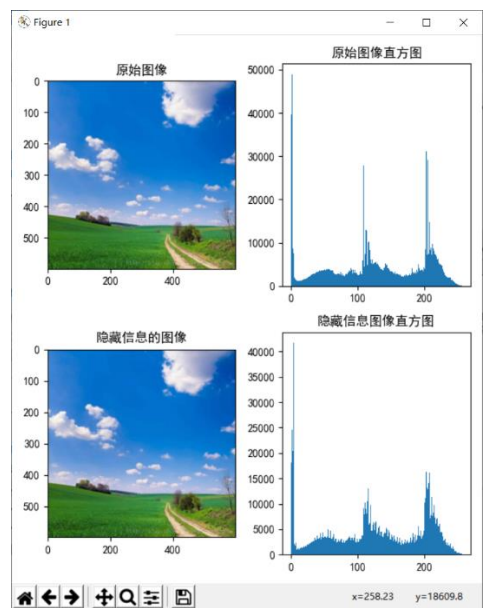


图 5.56 图像降级算法改进的原始图像和隐藏信息的图像对比图以及直方图

5.3.2 变换域水印

(一) DCT 水印嵌入

首先点击“DCT 算法水印嵌入”按钮，进行嵌入操作。



图 5.57 点击“DCT 算法水印嵌入”按钮

下面会弹出提示，请用户选择载体图像，选择图像后点击“打开”。



图 5.58 载体图像

下面会提示用户选择要隐藏的信息，用户需要提前将要隐藏的信息放入一个 txt 文本文件中，然后选择这个文件。

这是我们准备隐写入图像中的信息，

“Visual watermarking system based on digital image by hxx” 选择好信息文件后点击“打开”。

下面程序就会将隐藏信息用 DCT 算法，嵌入到用户选择的图片中。隐写成功后会弹出提示，告诉用户隐写已完成，并且将隐写后的图像保存在程序所在的目录下，命名为用户选择文件的文件名+_DCT-generated

点击确定后，会出现原始图像和隐藏信息的图像的对比图，并绘制它们的直方图。

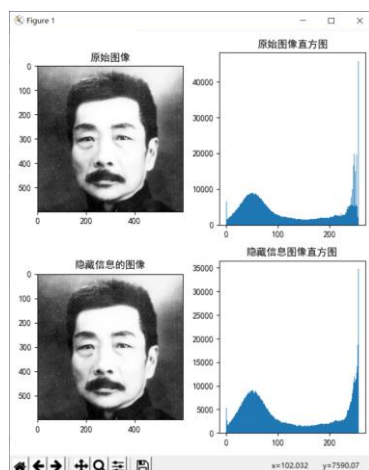


图 5.59 原始图像和隐藏信息图像的对比图以及直方图

2、DCT 水印提取

DCT 水印提取和 LSB 水印提取不同的一点是，DCT 水印提取输入的信息长度必须是正确的信息长度，否则将会是乱码，不能提取出正确的信息。

首先设置提取信息的长度，然后点击“DCT 水印提取”按钮。



图 5.60 点击“DCT 水印提取”按钮

按照弹出的提示，选择要提取隐藏信息的图像和要将提取的信息保存的位置，选择好保存的文件夹后点击“选择文件夹”。

之后弹出提示，隐藏信息已提取，保存为 DCT_hidden_text.txt。然后来到选择保存信息的文件夹，打开 DCT_hidden_text.txt，可以看到信息已被正确提取。

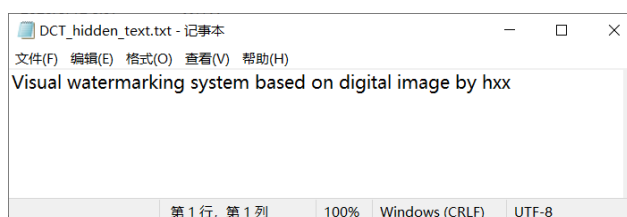


图 5.61 提取的隐藏信息

附录

表 1 系统运行程序和源代码

文件名	文件说明
main.exe	系统运行程序
main.py	系统源代码

表 2 系统运行使用的图片和文本文件

文件名	文件说明
隐写用图	
• LSB_origin.png	LSB 基本算法 用图
• DCT_origin.bmp	DCT 算法 用图
• random_interval_origin.bmp	随机间隔算法 用图
• regional_verification_origin.bmp	区域校验位算法 用图
• image.png	图像降级算法及其改进 载体图片
• mark.png	图像降级算法及其改进 水印图片
隐藏信息	
• hideInfo_LSB.txt	LSB 基本算法 水印文本
• hideInfo_DCT.txt	DCT 算法 水印文本
• hideInfo_random_interval.txt	随机间隔算法 水印文本
• hideInfo_regional_verification.txt	区域校验位算法 水印文本