



南昌大学实验报告

学生姓名: 丁俊 学 号: 8003119100 专业班级: 信息安全 193 班
实验类型: ☐ 验证 ☐ 综合 ☐ 设计 ☐ 创新 实验日期: 2022.3.4 实验成绩: _____

一、实验项目名称

LSB 图像隐写与提取算法

二、实验目的

了解信息隐藏中最常用的 LSB 算法特点, 掌握 LSB 算法原理, 设计并实现一种基于图像的 LSB 隐藏算法; 了解如何通过峰值信噪比来对图像质量进行客观评价, 并计算峰值信噪比。

三、实验基本原理

任何多媒体信息, 在数字化时, 都会产生物理随机噪, 而人的感观系统对这些随机噪声不敏感。信息隐藏技术就是利用这个原理, 通过使用秘密信息比特替换随机噪声, 从而完成信息隐藏目标。

BMP 灰度图像的每个像素值为 8bit 二进制值, 表示该点亮度。不同位平面对视觉影响不同, 图像高位平面对图像感官质量起主要作用, 去除图像最低位平面并不会造成画面质量的明显下降。利用这个原理可用秘密信息(或称水印信息)替代载体图像低位平面以实现信息嵌入。

本实验中算法选用最低位平面来嵌入秘密信息。最低位平面对图像的视觉效果影响最轻微, 但很容易受噪声影响和攻击, 解决办法可采用冗余嵌入的方式来增强稳健性。即在一个区域(多个像素)中嵌入相同的信息, 提取时根据该区域中的所有像素判断。

四、主要仪器设备及耗材

Window10、pycharm、pngRGB 图片

五、实验步骤

思路：我们先把要隐藏的信息内容转成二进制字符串，提取出目标图像的像素点的 RGB 三原色的十进制数值，将要隐藏的二进制 01 串分别隐藏到每一个 0~255 范围的 RGB 值的最后一位，利用 python 中的 PIL 的 putpixel() 函数生成隐藏信息图片。提取的时候反过来把得到的信息隐藏图片按照像素点逆提取出二进制数位按 8 位一组恢复成对应的 Ascii 数值。

LCB_hide.py - 隐写程序

```
1.  from PIL import Image
2.
3.  def mod(x, y):
4.      return x % y
5.
6.  # 获取隐藏文件的信息转成二进制子串
7.  def bin_ord(flag):
8.      string = ""
9.      with open(flag) as f:
10.         txt = f.read()
11.         for i in range(len(txt)):
12.             string = string + bin(ord(txt[i])).replace('0b', '').zfill(8)
13.         return string
14.
15.
16. def hide(pic, flag, new_pic):
17.     count = 0
18.     im = Image.open(pic)
19.     width = im.size[0]
20.     height = im.size[1]
21.     string = bin_ord(flag)
22.     for h in range(height):
23.         for w in range(width):
24.             pixel = im.getpixel((w, h))
25.             x = pixel[0]
26.             y = pixel[1]
27.             z = pixel[2]
28.             if count == len(string):
29.                 break
30.             x = x - mod(x, 2) + int(string[count])
31.             im.putpixel((w, h), (x, y, z))
32.             count = count + 1
33.             if count == len(string):
34.                 break;
35.             y = y - mod(y, 2) + int(string[count])
```

```

36.         im.putpixel((w, h), (x, y, z))
37.         count = count + 1
38.         if count == len(string):
39.             break
40.         z = z - mod(z, 2) + int(string[count])
41.         im.putpixel((w, h), (x, y, z))
42.         count = count + 1
43.     im.save(new_pic)
44.
45.
46. pic = 'main.png'
47. flag = 'dingjun.txt'
48. new_pic = 'new.png'
49. hide(pic, flag, new_pic)

```

LCB_Extract.py - 提取程序

```

1.     from PIL import Image
2.     def mod(x, y):
3.         return x % y
4.
5.     def extract(pic, lenth, hide):
6.         binary = ""
7.         string = ""
8.         count = 0
9.         im = Image.open(pic)
10.        width = im.size[0]
11.        height = im.size[1]
12.        for h in range(height):
13.            for w in range(width):
14.                pixel = im.getpixel((w, h))
15.                x = pixel[0]
16.                y = pixel[1]
17.                z = pixel[2]
18.                if count == lenth:
19.                    break
20.                binary = binary + str(mod(x, 2))
21.                count = count + 1
22.                if count == lenth:
23.                    break
24.                binary = binary + str(mod(y, 2))
25.                count = count + 1
26.                if count == lenth:
27.                    break

```

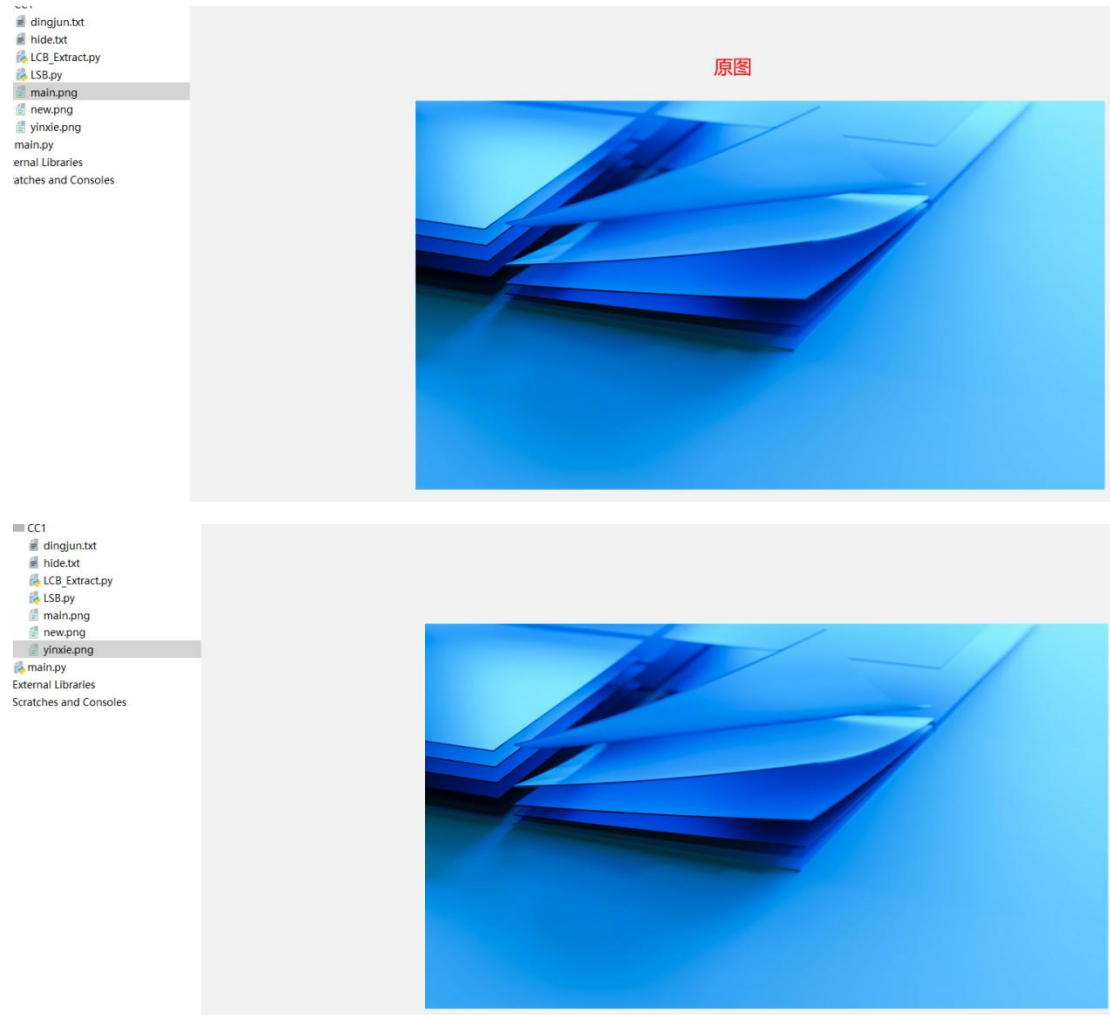
```

28.         binary = binary + str(mod(z, 2))
29.         count = count + 1
30.     with open(hide, "w", encoding="UTF-8") as f:
31.         for i in range(0, len(binary), 8):
32.             string = string + chr(int(binary[i:i + 8], 2))
33.         f.write(string)
34.
35. pic = 'new.png'
36. lenth = 200
37. hide = 'hide.txt'
38. extract(pic, lenth, hide)

```

六、实验数据及处理结果

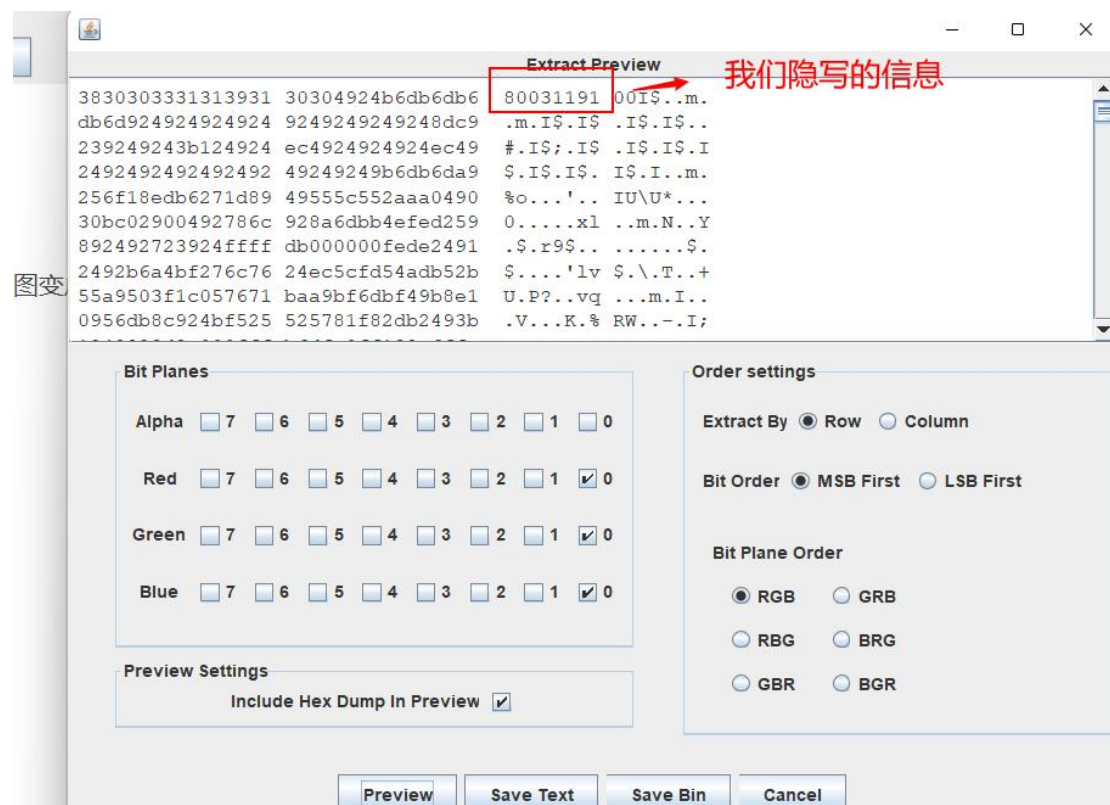
原图与隐写之后的图片



隐藏信息嵌入与提取



此外，我们还可以通过 StegSolve 对图片隐写的内容进行查看，如图。



七、思考讨论题或体会或对改进实验的建议

本实验对部分 jpg 格式文件可能无效，需要对 jpg 图片格式进行处理，而且如果可以使用 matlab 软件进行处理会更方便。

八、参考资料

信息隐藏技术