

总结

1. 本周学习情况

- 本周开始，阅读了基于位平面改进的图像信息隐藏算法，这个是在LSB算法的改进基础上进行的，对于LSB，隐藏位置有随机性，鲁棒性有所提高，抗击打能力增强，具有较高的使用价值，文中也通过实验证明了该算法的可行性。
- 收获：通过阅读该论文，觉得收获最大的不是知晓LSB的改进算法，而是知道了，对论文的编写，应该是先对改进的算法进行分析，哪里做的不好，哪些问题需要改正，简单的叙述其不足，再在这个基础上提出自己的方法和实验验证以及对比评价。
- 本周后半段，阅读了空域算法的相关技术文章(CSDN、知乎)，通过代码简单实现了图像线性融合算法和LSB算法
 - 代码贴图

```
LSB.py x
42     return added_image
43
44
45 def LSB_encoder(copyright_image, original_image):
46     """
47         使用LSB添加水印图片到原始图像中
48     :param copyright_image: numpy类型的RGB图片
49     :param original_image: numpy类型的RGB图片
50     :return: 一张使用LSB算法得到的图片
51     """
52     # 1. 确保输入图像为8bits无符号整型
53     original_image = original_image.astype(np.uint8)
54     copyright_image = copyright_image.astype(np.uint8)
55
56     # 2. 对图像进行备份，不能在原图上更改
57     water_mark = original_image.copy()
58     copy_right = copyright_image.copy()
59
60     # 3. 将copyright二值化，使其仅含0和1，用1bit表示
61     copy_right[copy_right < 200] = 1
62     copy_right[copy_right >= 200] = 0
63
64     # 4. 将watermark的最后1bit的R、G、B全部置零
65     # 也可以仅仅对R通道置零
66     for i in range(0, water_mark.shape[0]):
67         for j in range(0, water_mark.shape[1]):
68             # 这里整除2再乘上2就是对最后一位bit置零
69             """
70             示例：
71             175 -> 10101111    175 // 2 = 87 -> 87 * 2 = 174
72             174 -> 10101110
73             """
74
75     LSB_encoder()
```

```

16
17 # 图像线性融合添加水印
18 def image_add(copyright_image, original_image, alpha):
19     """
20     此函数是将水印图片加到原始图片中
21     :param copyright_image: RGB image 必须是 numpy type
22     :param original_image: RGB image 必须是 numpy type
23     :param alpha: 将水印图片加到原始图片的权重大小
24     :return: 最终添加后的图片
25     """
26     # 1. 确认输入的版权图片长度和宽度必须小于原始图片
27     if copyright_image.shape[0] > original_image.shape[0] or \
28         copyright_image.shape[1] > original_image.shape[1]:
29         raise ValueError('输入的版权图片长度和宽度必须小于原始图片')
30
31     original_image = original_image.astype(np.float32)
32     copyright_image = copyright_image.astype(np.float32)
33
34     # 2. 获取原始图片的备份, 计算中不能使用原始图片进行像素的计算
35
36     added_image = original_image.copy()
37
38     # 3. 按照alpha作为权重实现加法
39     h, w = copyright_image.shape[0], copyright_image.shape[1]
40     added_image[0:h, 0:w, :] = alpha * copyright_image + (1 - alpha) * original_image
41
42     return added_image

```

○ 结果展示

■ 使用LSB对图片添加水印

