

FFT图片隐写术

——基于FFT和私钥加密技术的数字图像水印系统

孙艺 李昊峻 曾文博 马梓豪

2023级信息工程创新班



工程背景

传统水印影响美观



传统水印易被抹除引发所有权纠纷



图片类知识产权侵权案件激增的司法应对

作者：曲阜市人民法院 发布时间：2023年12月21日

图片类知识产权侵权案件激增的司法应对

近年来，进入司法领域的图片类知识产权侵权案件不断增多，部分公司在图片版权取得、管理、授权过程中的乱象饱受社会公众质疑，批量化商业维权的模式已经占据主导地位，一定程度上造成司法定价替代正常市场行为，进而滋生不良的财富积累途径，破坏法治化营商环境。对此，人民法院应分类施策，积极应对，统一裁判尺度，有效抑制图片类侵权案件商业维权带来的非正常案件增长现象。

工程概述

技术名称： FFT图片隐写术

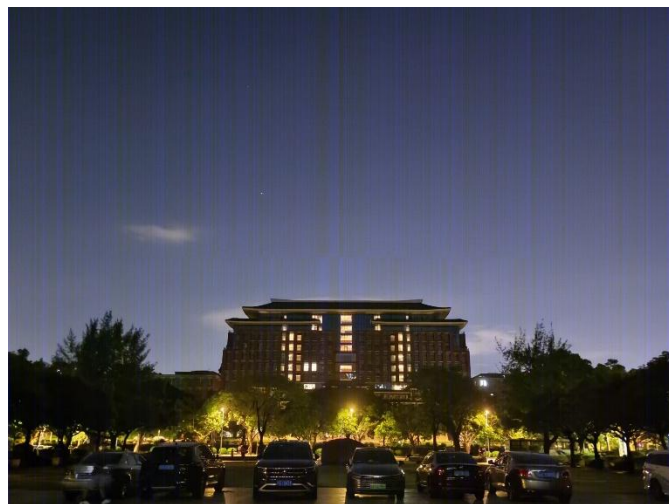
技术原理： 基于快速傅里叶变换技术（FFT），通过频域处理实现数字水印的嵌入与提取，并通过私钥控制的随机映射机制，确保水印嵌入位置的安全性。

技术作用： 能够对图片进行水印嵌入、提取和恢复操作，支持使用私钥保护水印。

实例：



=



理论分析

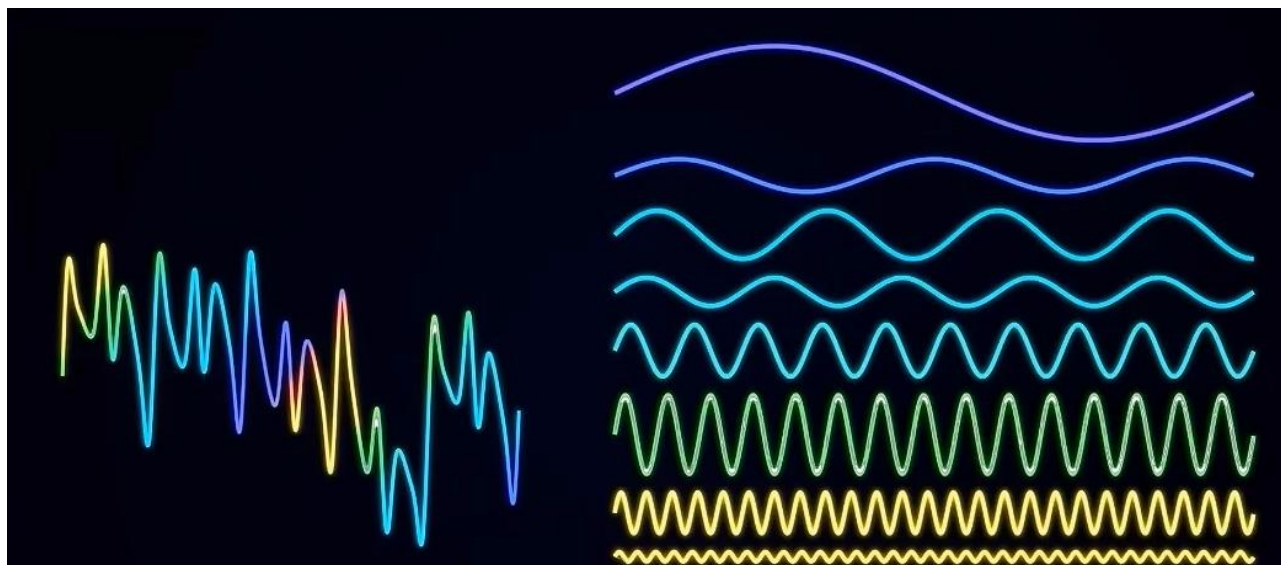
DSP理论基础

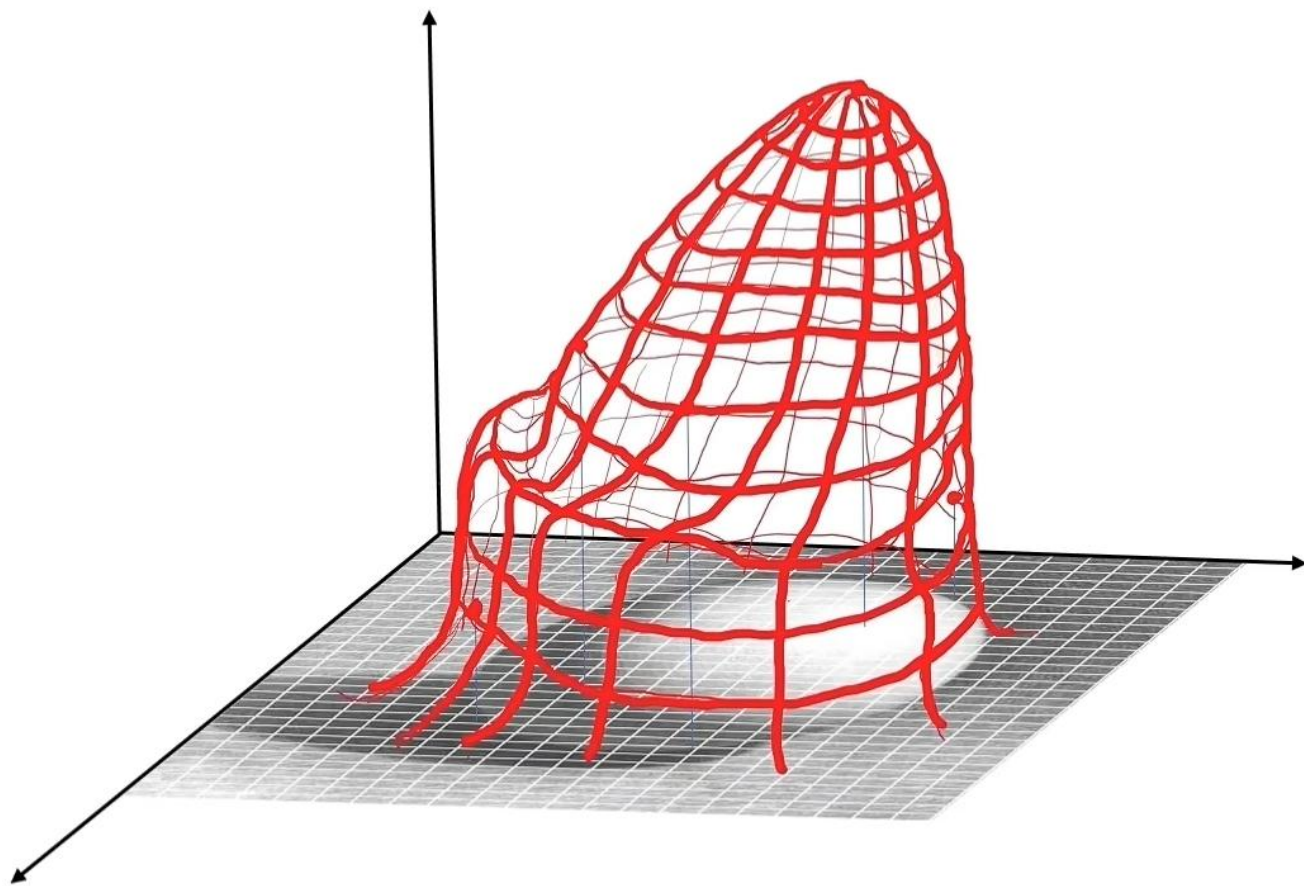
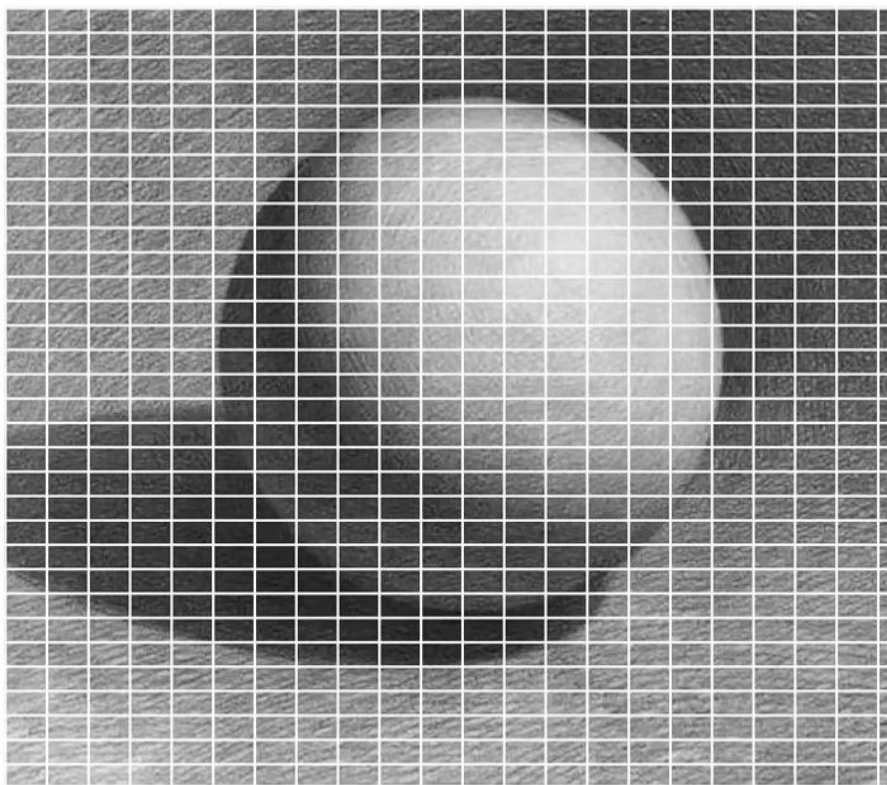
傅里叶变换是信号处理的核心工具，它将时域信号转换到频域。对于图像，二维FFT将空域的像素矩阵转换为频域的频谱矩阵，揭示图像的频率成分。

一维连续傅里叶变换及反变换

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx$$

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du$$





二维离散傅里叶变换

设 $f(x,y)$ 是一个 $M \times N$ 的图像，其中 $x=0,1,\dots,M-1$ 和 $y=0,1,\dots,N-1$ 。则其二维离散傅里叶变换 $F(u,v)$ 定义为：

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$$

其中， $u=0,1,\dots,M-1$ 和 $v=0,1,\dots,N-1$ 。

二维离散傅里叶逆变换：

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$$

频域特性

频域有三个关键特性。

第一，共轭对称性保证实数图像变换后可逆；

第二，不同的频率对应不同的图像信息；

第三，频域修改对应空域全局变化，这是水印不可见的理论基础。

共轭对称性：对于实数信号： $F(u,v) = F^*(-u,-v)$

幅度谱： $|F(u,v)| = |F(-u,-v)|$ （偶对称）

相位谱： $\angle F(u,v) = -\angle F(-u,-v)$ （奇对称）

频率成分：

低频：图像的主要内容（轮廓、亮度）

中频：纹理细节

高频：边缘、噪声

私钥加密机制

为什么需要私钥？

安全性：没有密钥无法提取水印

防篡改：错误密钥导致失真，可检测篡改

灵活性：支持任意字符串作为密钥

私钥工作原理

1. 字符串转数值

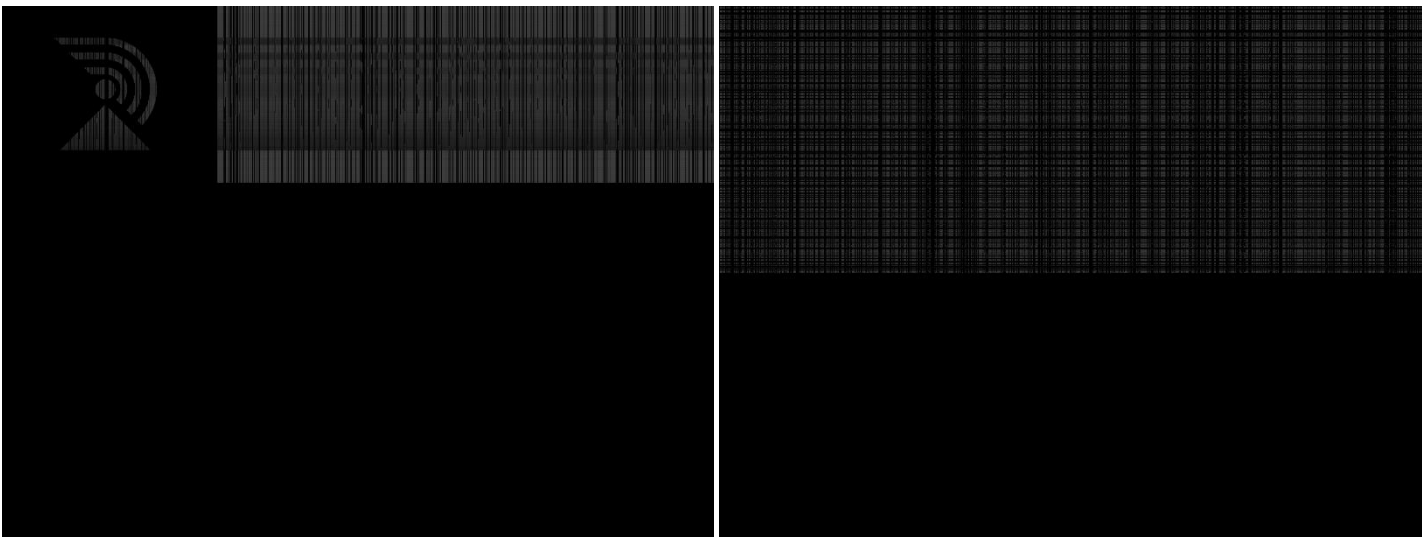
```
seed = sum(ord(c) * (i + 1)
            for i, c in enumerate(key_str))
```

2. 生成随机位置映射

```
random.seed(seed)
random.shuffle(x_positions)
random.shuffle(y_positions)
```

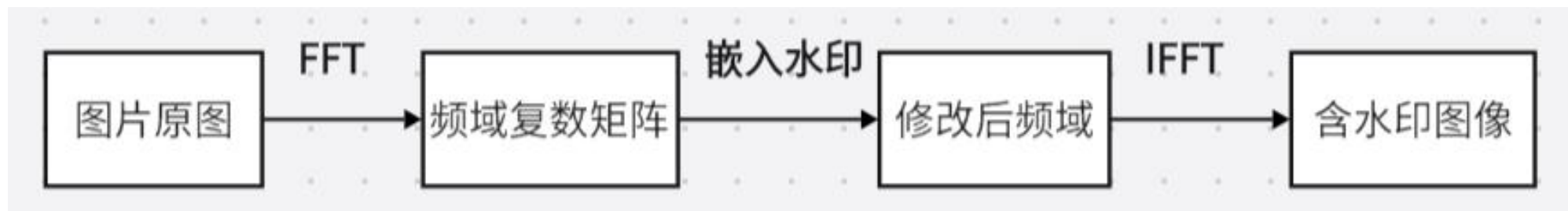
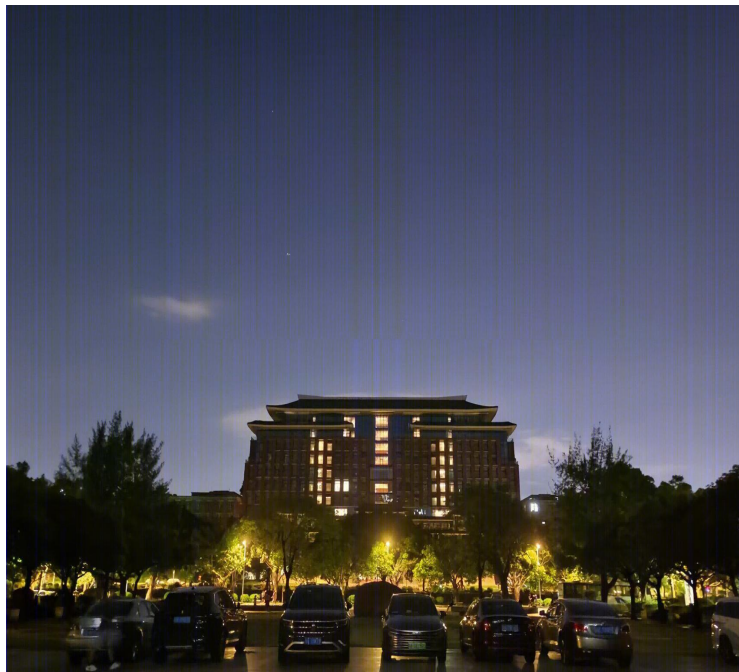
3. 按映射嵌入水印

```
for i in range(height // 2):
    for j in range(width):
        if x[i] < wm_height and y[j] < wm_width:
            temp[i][j] = watermark[x[i]][y[j]]
            temp[height-1-i][width-1-j] = temp[i][j]
# 共轭对称
```



水印嵌入原理

水印嵌入的核心思想是在频域叠加水印信号。通过控制嵌入强度 α ，平衡不可见性和鲁棒性。由于频域的对称性，我们需要对称嵌入水印，确保逆变换后仍是实数图像。



准备水印数据并嵌入频谱

```
def prepare_watermark(img_fft, watermark, alpha, private_key=None):  
  
    height, width, channel = img_fft.shape  
    wm_height, wm_width = watermark.shape[0], watermark.shape[1]  
    x, y = list(range(height // 2)), list(range(width))  
    seed = private_key if private_key is not None else (height + width)  
    random.seed(seed)  
    random.shuffle(x)  
    random.shuffle(y)  
    temp = np.zeros(img_fft.shape)  
    for i in range(height // 2):  
        for j in range(width):  
            if x[i] < wm_height and y[j] < wm_width:  
                temp[i][j] = watermark[x[i]][y[j]]  
                temp[height - 1 - i][width - 1 - j] = temp[i][j]  
    return img_fft + alpha * temp
```

param img_fft: 图片的频谱数据

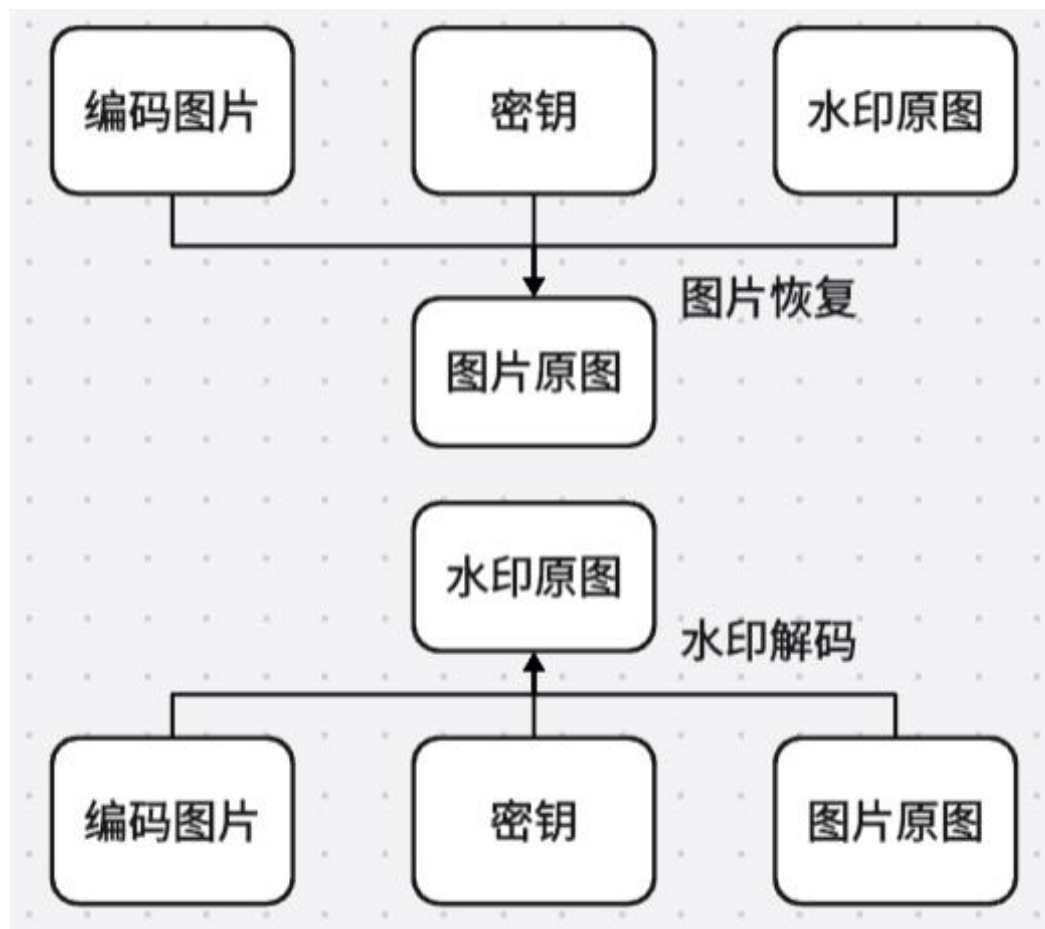
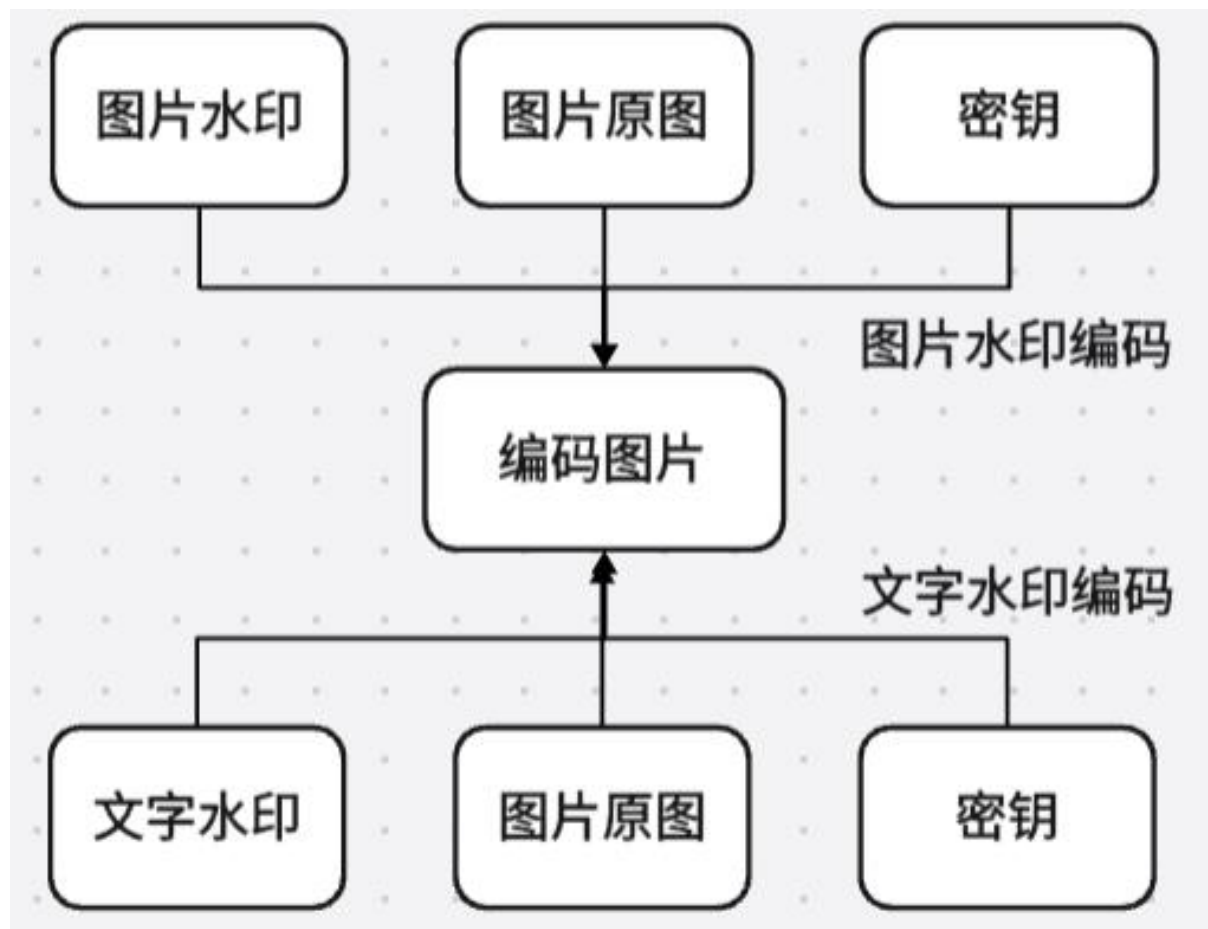
param watermark: 水印图片

param alpha: 水印强度

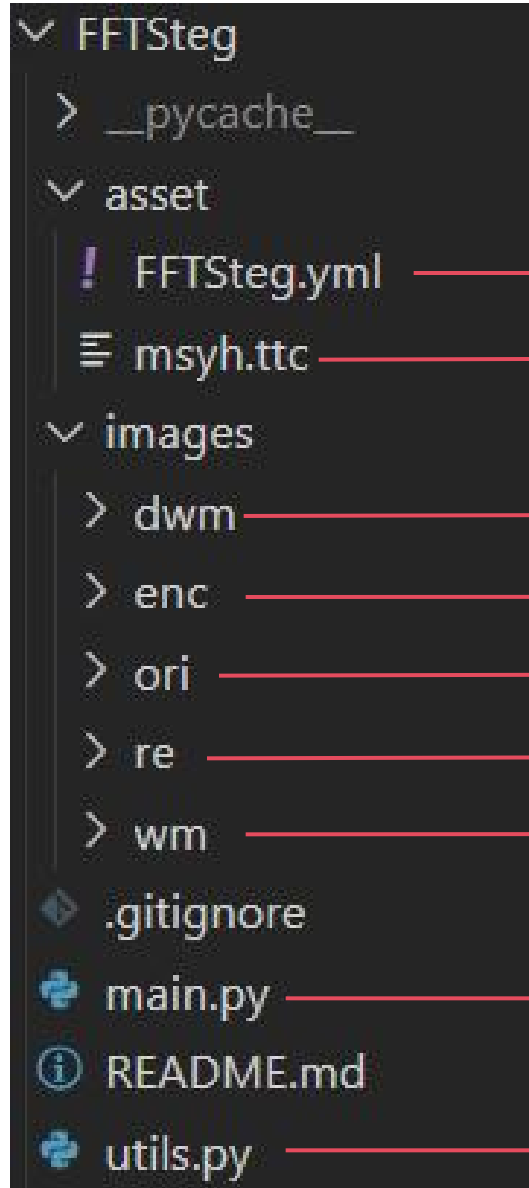
param private_key: 私钥, 用于加密水印位置, 如果为None则使用图像尺寸作为种子

return: 嵌入水印后的频谱数据

文件流程图



🔍 文件架构



配置文件

微软雅黑字体

解码水印

编码图片

原图

恢复后图片

水印图

主函数

库



代码演示

项目总结

- **核心技术：** 基于二维FFT和私钥加密技术的频域水印
- **核心成果：**
 - 实现了**不可见**的水印嵌入
 - 引入了**私钥加密**机制，增强安全性
 - 系统具备一定的**鲁棒性**
- **项目价值：** 深化了对DSP理论的理解，并完成了实践应用

未来展望

- **增强鲁棒性：** 对抗更复杂的图像处理攻击
- **智能自适应：** 根据图像内容动态调整水印强度
- **拓展应用领域：** 从静态图像保护迈向视频、音频等领域

参考文献

- [1]Red Car. Python3简单实现隐写术 [EB/OL].(2023-3-22)[2025-10-13].http://blog.csdn.net/m0_67373485/article/details/129702142
- [2]曲阜市人民法院.图片类知识产权侵权案件激增的司法应对[EB/OL]. (2023-12-21) [2025-10-11]. <http://jningqffy.sdcourt.gov.cn/jningqffy/385451/385457/14635834/index.html>
- [3]孙海英.一种基于傅里叶变换的数字音频水印仿真实现[J].科技信息,2008,(36):77-78+133.DOI:CNKI:SUN:KJXX.0.2008-36-056.

FFTSteg

本项目已在GitHub开源：<https://github.com/Siryhahaha/FFTSteg>

感谢各位的聆听，欢迎各位点亮Star 🌟

