



Project 2: 基于数字水印的图片泄露检测

学院: 网络空间安全学院

专业: 密码科学与技术

姓名: 李双平

学号: 202200180026

2025 年 8 月 15 日

目录

1 水印嵌入与提取	2
1.1 实现原理	2
1.2 代码解释	2
1.3 实验结果	4

1 水印嵌入与提取

1.1 实现原理

本实验基于 **LSB 数字水印** 技术，在灰度图像的最低有效位嵌入二值水印信息。LSB 方法的核心思想是：

- 灰度图像的每个像素值由 8 位二进制组成，其中最低有效位对整体亮度影响极小；
- 通过替换最低有效位为水印像素的值，可以在不明显影响宿主图像视觉效果的情况下完成水印嵌入；
- 提取水印时，只需读取最低有效位即可恢复原水印信息。

在鲁棒性测试部分，实验对含水印图像进行了多种常见图像处理（如翻转、平移、旋转、缩放、调整对比度等），并计算水印提取结果与原水印之间的相似度，用于评估嵌入方法的稳定性。

1.2 代码解释

以下代码使用 Python + OpenCV 实现了 LSB 嵌入、提取以及鲁棒性测试。

1. LSB 水印嵌入函数

```
1 def lsb_insert(host_img, mark_img):  
2     mark_resized = cv2.resize(mark_img, (host_img.shape[1], host_img.shape  
3         [0]))  
4     mark_bin = (mark_resized > 127).astype(np.uint8)  
5  
6     merged_img = host_img.copy()  
7     merged_img = (merged_img & 0b11111110) | mark_bin  
8  
9     return merged_img
```

功能：将二值水印嵌入灰度宿主图像的最低有效位 (LSB)。实现步骤：

- 调整水印大小，使其与宿主图像一致；
- 将水印二值化 (0/1)；
- 将宿主图像每个像素的最低有效位清零；
- 用水印像素替换最低有效位，得到嵌入水印后的图像。

2. LSB 水印提取函数

```
1 def lsb_extract(img_with_mark):  
2     return img_with_mark & 1
```

功能：从最低有效位提取水印。实现原理：通过按位与操作 (`& 1`) 获取图像中每个像素的最低有效位，得到提取出的二值水印。

3. 鲁棒性评估函数

```
1 def evaluate_resilience(img_with_mark, orig_mark):
2     ops = {
3         "水平翻转": lambda x: cv2.flip(x, 1),
4         "平移": lambda x: cv2.warpAffine(x,
5             np.float32([[1, 0, 30], [0, 1, 30]]),
6             (x.shape[1], x.shape[0])),
7         "旋转90度": lambda x: cv2.rotate(x, cv2.ROTATE_90_CLOCKWISE),
8         "缩小一半": lambda x: cv2.resize(x, (x.shape[1] // 2, x.shape[0] //
9             2)),
10        "增加对比度": lambda x: cv2.convertScaleAbs(x, alpha=1.4, beta=0)
11    }
12
13    for op_name, func in ops.items():
14        transformed = func(img_with_mark)
15        extracted = lsb_extract(transformed)
16        similarity = np.mean(extracted == (orig_mark > 127).astype(np.
17            uint8))
18        print(f"{op_name} 处理后相似度: {similarity * 100:.2f}%")
```

功能：评估水印在不同图像处理操作后的相似度。主要步骤：

- 定义多种图像变换操作（翻转、平移、旋转、缩放、调对比度等）；
- 对含水印图像进行变换后提取水印；
- 与原始水印对比，计算相似度（匹配像素比例）。

4. 主程序

```
1 if __name__ == "__main__":
2     host = cv2.imread("original_image.jpg", cv2.IMREAD_GRAYSCALE)
3     mark = cv2.imread("watermark.jpg", cv2.IMREAD_GRAYSCALE)
4
5     if host is None or mark is None:
6         print("请确保 original_image.jpg 和 watermark.jpg 存在并放在代码目
7             录下")
8         exit()
9
10    watermarked = lsb_insert(host, mark)
11    cv2.imwrite("watermarked.jpg", watermarked)
12
13    recovered_mark = lsb_extract(watermarked) * 255
14    cv2.imwrite("extracted_watermark.jpg", recovered_mark)
```

```
15     print("\n开始鲁棒性测试: ")  
16     evaluate_resilience(watermarked, mark)
```

功能：

- 读取宿主图像与水印图像；
- 调用 `lsb_insert` 进行水印嵌入，并保存结果；
- 调用 `lsb_extract` 提取水印并保存；
- 调用 `evaluate_resilience` 进行鲁棒性评估。

1.3 实验结果

实验结果如下：



图 1：原图

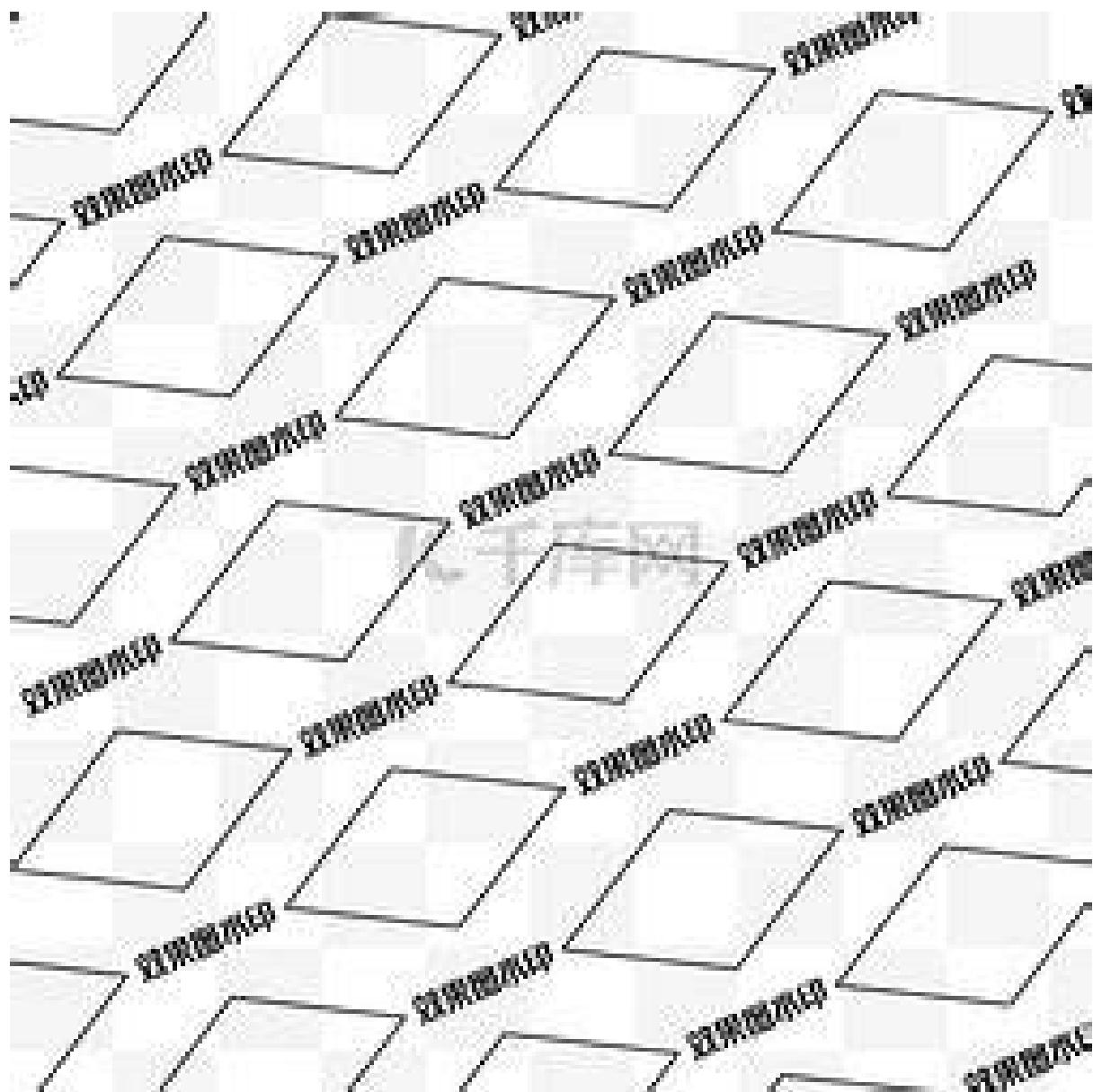


图 2: 水印

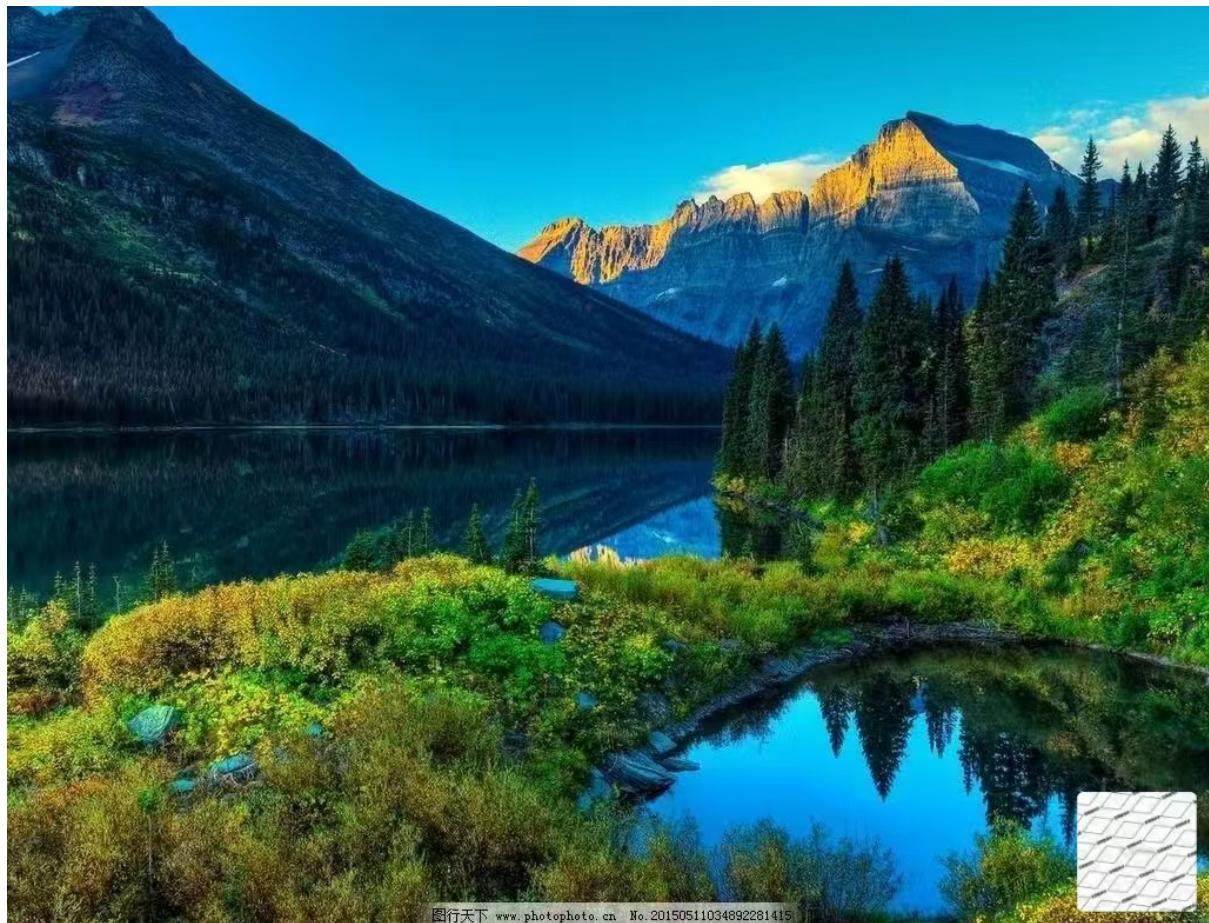


图 3: 嵌入水印的图