

两阶段可分离深度学习框架for实用盲水印

[A-Novel-Two-stage-Separable-Deep-Learning-Framework-for-Practical-Blind-Watermarking/model.py at master · MengxiGuo/A-Novel-Two-stage-Separable-Deep-Learning-Framework-for-Practical-Blind-Watermarking \(github.com\)](https://github.com/MengxiGuo/A-Novel-Two-stage-Separable-Deep-Learning-Framework-for-Practical-Blind-Watermarking/blob/master/model.py)

没有训练和测试代码

OET one-stage end-to-end training 一阶段端到端训练

缺陷：噪声攻击必须以可微分的方式进行模拟，这在实践中并不总是适用。经常遇到收敛缓慢的问题，并且在噪声攻击下往往会降低水印图像的质量。

TSDL框架 two-stage separable deep learning

- **FEAT** noise-free end-to-end adversary training 无噪声端到端对抗训练
开发了冗余多层特征编码网络来获得编码器
- **ADOT** noise-aware decoder-only training 仅噪声感知解码器训练
获得足够鲁棒且实用的解码器，可以接受任何类型的噪声。

Introduction

盲水印技术——应对多图像抠图算法

盲水印技术属性

- 水印图像的质量——保证水印不易察觉
- 水印的鲁棒性——引导水印在各种噪声攻击中幸存下来

传统的盲水印方法

- 时域法
- 频域法

OET

它由编码器、噪声层和解码器三个组件组成。

- 编码器用于为输入图像添加水印
- 噪声层模拟对水印图像的噪声攻击
- 解码器负责从嘈杂的图像中恢复水印

请注意，恢复的水印和输入水印之间的偏差会通过所有三个组件传播回来，并且编码器和解码器的参数以端到端的方式同时训练。

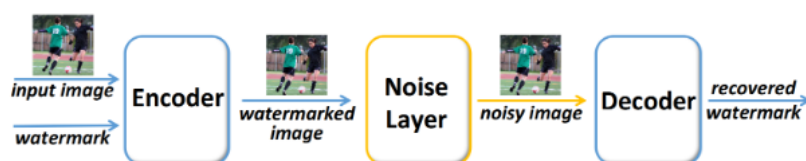


Figure 2: A typical One-stage End-to-end Training (OET) deep network architecture.

局限性：

- 编码器和解码器必须使用可微分噪声层进行训练，这意味着噪声必须支持反向传播。（实践中不适用）

- 每当引入新型噪声时，在再训练过程中需要同时调整编码器和解码器的所有参数，这在计算上是相当昂贵的
- 一旦显示带水印的图像，就无法调用原始图像进行重新处理。因此，重新训练整个模型很难挽回损失
- 对超参数非常敏感，因为需要联合训练多个组件。损失函数始终收敛于降低水印图像质量的方向，以保证训练过程中的解码精度

TSDL

1. 第一阶段，TSDL采用多层特征编码策略来获取一个强大的编码器，该编码器可以在不参考任何噪声的情况下自主地将水印信息冗余编码到输入图像中，我们称之为无噪声端到端对手训练（FEAT）编码器在没有看到任何噪音的情形下学习抗噪水印模式，提出RMFEN（redundant multi-level feature encoding network）来作为框架，该框架涉及多层图像特征，用于水印信息的冗余共编码。
2. 第二阶段，称为噪声感知解码器仅训练（ADOT），编码器的参数不再修改，但解码器会根据来自第一阶段的预训练解码器的不同噪声进行微调。

引入强度因子来控制鲁棒性和不可感知性之间的权衡。

测量常见传统噪声和黑盒噪声攻击下的鲁棒性来分析该方法的性能

Related Work

数字水印

方式：

- 最低有效位LSB嵌入
- 频域水印，如DCT域、DFT域和DFT域

应用：图像、视频、音频和其他多媒体文档的版权保护

用于水印的深度学习

CNN（卷积神经网络）应用于水印，其非盲水印比传统方法具有更好的隐蔽性和鲁棒性【22】

基于CNN的盲水印架构，并使用相同的网络进行水印的嵌入和提取【30、31】

将对抗网络引入盲水印的研究，在空间域中对水印进行了编码【42】

引入了残差（residual）和在变换域（transform domain）中嵌入水印的思想，在没有对抗网络的情况下实现了出色的鲁棒性和高质量的图像【1】

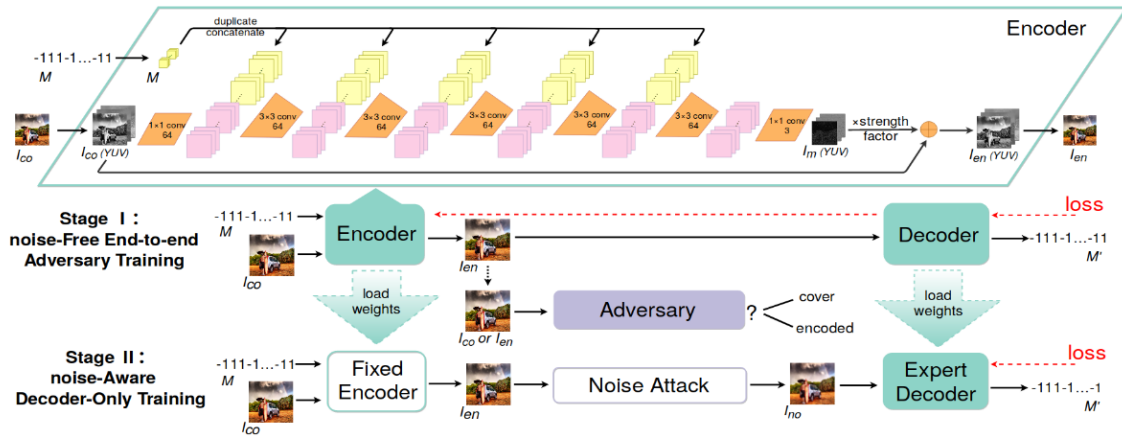
对抗网络

将对抗训练引入生成对抗网络（GAN）中，用于估计生成模型【15】

应用频谱归一化使对抗网络收敛更加稳定【29】

Proposed Framework

模型架构



• 编码器 E , 参数 θ_E

- 输入: 表面图片 cover image $I_{co} \in \mathbb{R}^{C \times H \times W}$, 秘密信息 $M \in \{-1, 1\}^L$
- 输出: 已编码的图片 $I_{en} \in \mathbb{R}^{C \times H \times W}$
- 具体编码过程:
 1. 消息 M 通过复制扩展为 3 维张量 $\{-1\}^{L \times H \times W}$
 2. 将复制的 3D 消息张量连接到每层的输出特征 $\in \mathbb{R}^{64 \times H \times W}$ 中, 并将新的张量 $\in \mathbb{R}^{(64+L) \times H \times W}$ 输入到下一层里, 充分利用不同级别的特征来整合水印信息, 得到带有消息 M 的水印掩码 I_m 。
 3. 编码图像 $I_{en} = I_{co} + S * I_m$, S 是强度因子, 用于控制嵌入水印的强度。
 4. 损失函数

$$\mathcal{L}_E = MSE(I_{co}, I_{en}) = MSE(I_{co}, E(\theta_E, I_{co}, M))$$

• 噪音攻击部分

- 输入: I_{en}
- 输出: 使用嘈杂图像 I_{no} 的输出对输入进行破坏

• 解码器 D , 参数 θ_D

- 输入: I_{en} 或 I_{no}
- 输出: 恢复的信息 M'
- 一些细节:

1. $M \in \{-1, 1\}^L$ 而不是 $M \in \{0, 1\}^L$ 的原因:

$M \in \{-1, 1\}^L$ 时, 带有水印的图像的分布趋向 -1 和 1, 没有水印的图像分布趋向于 0

Q: 解码器的训练样本是没有不带水印的图像吗?

2. 损失函数

$$\mathcal{L}_D = MSE(M, M') = MSE(M, D(\theta_D, \tilde{I})), \tilde{I} \in \{I_{en}, I_{no}\}$$

• 对手 A , 参数 θ_A

- 输入: I_{en} 或 I_{no}
- 输出: 给定图像是编码图像的概率
- 一些细节:

1. 损失函数 (用于通过更新 θ_E 来提高 I_{en} 的视觉质量)

$$\mathcal{L}_A = \log(1 - A(I_{en})) = \log(1 - A(E(\theta_E, I_{co}, M)))$$

2. 对抗训练是通过最小化**值函数**和更新参数 θ_A 来实现的:

$$\mathcal{L}_2 = \mathcal{V}(E, A) = \log(1 - A(\theta_A, I_{co})) + \log(A(\theta_A, E(I_{co}, M)))$$

两阶段可分离训练

第一阶段: 无噪声端到端对手训练 FEAT

将已编码的图像直接输入进解码器, 训练目标是最小化:

$$\mathcal{L}_1 = \lambda_E \mathcal{L}_E + \mathcal{L}_D + \lambda_A \mathcal{L}_A$$

λ_E 和 λ_A 是权重因子, 对手也参与了这一阶段

为了提高训练稳定性, 使用了谱归一化 (spectral normalization)

[Spectral Normalization 谱归一化 - 知乎\(zhihu.com\)](#)

FEAT的主要目标是获得一个强大的冗余编码编码器, 该编码器将在下一阶段**固定且保持不变**

第二阶段: 针对各种噪声的噪声感知仅解码器训练 ADOT

- 在此阶段, 引入噪声处理以有针对性地训练解码器
- 只关注神经网络中的解码器, 仅更新 θ_D 以最小化 \mathcal{L}_D
- 将从阶段 I 获得的解码器权重加载为预训练权重可以显著加速 ADOT
- 目标是充分利用解码器的潜力

传统的噪音攻击

主要是指一些典型的噪音

[数字图像学笔记——6. 噪音生成（椒盐噪音、高斯噪音、泊松噪音）泊松噪声打码的老程的博客-CSDN 博客](#)

- Resize noise 将编码后的图像缩小到 $(p * H, p * W)$, $p \in (0, 1)$, 然后再将图像缩放到原来的大小
- salt and pepper noise 随机让某些像素点变为 0 或 255
- Dropout 比率为 p 的像素点会被 cover image 相应位置图像随机替换
- Croppot and Crop 随机选择一片方形区域 $(\sqrt{p} * H, \sqrt{p} * W)$, $p \in (0, 1)$, 区域中像素不变, 区域外像素被 cover image 替换
- Gaussian blur noise 使宽度为 r 的高斯核的编码图像模糊
- JPEG 不可微分, 但本文训练方法可以直接引入

黑盒噪音攻击

主要是指日常生活中常见的图像处理软件引起的噪声攻击

使用图像批处理软件, 选择5种处理类型作为黑盒噪声攻击, 包括4种滤镜（星光、彩铅、蜡笔、铅笔素描）和1种可感知水印。彩铅涉及图像的不规则裁剪。可感知水印添加可见的噪点水印（透明度为0-100）。

Experiments

数据集

来自COCO数据集的10000张随机图像和来自CIFAR-10的996张图像

训练期间在1000张看不见的COCO图像上进行实验

在针对黑盒噪声的实验中，随机选择1100张来自COCO的图像并合成了上述提到的5种噪声，对每种噪声的数据集都包括1000张用于训练的图像和100张用于测试的图像

实现细节

使用PyTorch实现框架

参数设置

图像都被转换为大小为 $C * H * W = 3 * 128 * 128$ 的 YUV 空间

随机消息 M 的长度 L 为 30

权重因子 $\lambda_E = 0.7, \lambda_A = 0.001$

对于梯度下降，Adam 学习率为 10^{-4} ，默认为超参数

每个模型跑200个epoch，批量大小为12（这里的batch是批量处理黑盒噪声里的批量还是使用不同的传统噪声攻击的批量）

强度因子 S 训练期间设置为1，在测试期间分配不同的值

对于特定训练，训了20个特定解码器，5个针对黑盒噪声，15个针对8种不同强度的传统噪声。对于组合训练，只训练了一个组合解码器，对每个小批量使用不同的传统噪声攻击。**（这是说一个批量一个攻击还是一个批量多个攻击）**

[如何理解 YUV ? - 知乎 \(zhihu.com\)](#)

[一文看懂各种神经网络优化算法：从梯度下降到Adam方法 - 知乎 \(zhihu.com\)](#)

评价指标

PSNR 和 bit accuracy

- PSNR 测量 encoded image 和 cover image 之间的相似性，这表明编码图像的质量
- 鲁棒性是使用比特精度来衡量的，比特精度是输入消息 M 和输出消息 M' 之间相同比特数与总比特的比值

baseline

[1] Mahdi Ahmadi, Alireza Norouzi, S. M. Reza Soroushmehr, Nader Karimi, Kayvan Najarian, Shadrokh Samavi, and Ali Emami. 2018. ReDMark: Framework for Residual Diffusion Watermarking on Deep Networks. CoRR abs/1810.07248 (2018).

[42] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. 2018. HiDDeN: Hiding Data With Deep Networks. In ECCV.

PS：一个没开源，一个开源了但是没开源权重，按着论文的做但是做不出论文说的效果，只能直接和论文说的效果比较，很有意思啊很有意思

Quantitative Results 定量效果

通过更改强度因子 S 的值来调整模型图像质量和鲁棒性

定义了一个鲁棒性值 R_s ，为组合解码器在一定强度因子 S 下经过 23 种噪声测试后的平均比特精度， I 代表测试图像的数量， N 代表噪声类型的数量， L 代表消息 M 的长度， \odot 为逐元素相乘

$$R_s = \frac{\sum_{i=1}^{I=1000} \sum_{n=1}^{N=23} \sum_{l=1}^{L=30} M_{i,n,l,s} \odot M'_{i,n,l,s}}{I \times N \times L}$$

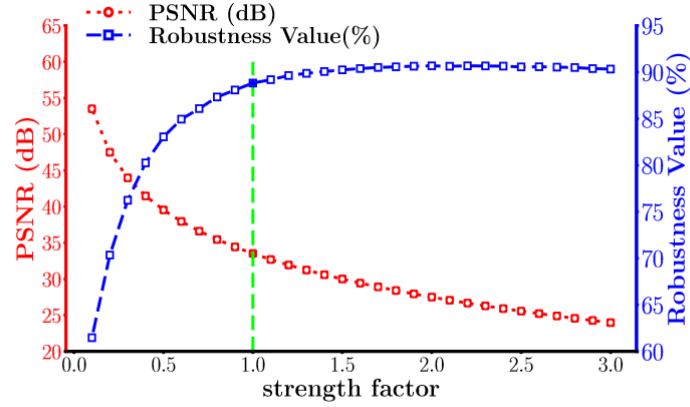


Figure 5: The curves of PSNR and Robustness Value varying with strength factor. There is an apparent trade-off between PSNR and robustness.

上述实验结果的解释：水印mask I_m 相当于一个扰动很小的噪点层，添加到图像中后会降低图像质量

Qualitative Results 定性结果 (S = 1)

- 传统噪音攻击

	Dropout ($p = 30\%$)	Cropout ($p = 9\%$)	Crop ($p = 1\%$)	Gaussian blur ($r = 3$)	JPEG ($Q = 50$)	Resize ($p = 50\%$)	Gaussian noise ($\sigma = 25$)	Salt&Pepper ($p = 10\%$)
I_{en}								
I_{no}								
$ I_{en} - I_{no} $								
PSNR(dB)	34.86	33.67	5.9	23.56	20.60	23.41	20.18	15.88
Pretrained(%)	90.6	67	62.4	87.4	54	73.7	58.75	55.6
Specified(%)	99	96.5	86.3	99.7	77.7	91.7	79.3	99.8
Combined(%)	97.5	91.9	75.3	99.1	76.2	92.8	74.4	90.9

Figure 4: Robustness against different traditional noises. Top: encoded image I_{en} ; Middle: noisy image I_{no} ; Bottom: magnified difference $|I_{en} - I_{no}|$. PSNR(dB) reflects the similarity between I_{en} and I_{no} . Pretrained(%) shows accuracy after Stage I where the decoder is trained without any noise. Specified(%) and Combined(%) respectively mean the accuracy of specified decoders and the combined decoder which are trained under an identity noise and multiple noises.

上图是预训练（进行了阶段I）、专用解码器与组合解码器的实验结果，8 种不同的专用解码器使用 8 种典型的高强度噪声进行训练

这充分体现了编码器的编码冗余，即使在这些高强度噪声失真下，编码图像也为解码器提供了足够的信息。

除了这 8 种类型之外，我们还在不同强度的噪声下训练了另外 7 个专门的解码器。

- 黑盒噪音攻击
















	Starlight	Crayon	Colored Pencil	Pencil Sketch	Watermarking ($V = 80\%$)
PSNR(dB)	21.3	12.55	7.91	10.66	10.88
Pretrained(%)	99	91	79.8	65.2	63.4
Specified(%)	—	99	98.2	86.4	94.7
I_{co}					
I_{en}					
I_{no}					

Figure 6: Robustness against black-box noises. We use an image batch-processing software, and select five types of processing as black-box noise attacks including four types of stylization and one type of Perceptible Watermarking. PSNR means the similarity between I_{en} and I_{no} . Pretrained decoder has the ability of resistance to Starlight and Crayon filter. For Colored Pencil, Pencil Sketch and Watermarking filter, satisfactory accuracy results are obtained after the specified training.

解决的问题：对于端到端的训练方法，黑盒噪音（封装的图像处理）无法模拟，因此在训练时没有办法引入黑盒噪音

OET 和 ADOT 之间的比较

在预训练模型的基础上，在指定噪声下进行 OET 和 ADOT 的比较。在这个实验中，我们选择cropout ($p = 9\%$) 作为噪声攻击。OET 的损失函数类似于阶段I 的 \mathcal{L}_1 损失。

- 图像质量的优势





			
I_{co}	OET $I_{en} (S = 1)$	ADOT $I_{en} (S = 1)$	ADOT $I_{en} (S = 1.5)$
PSNR(dB)	24.11	33.51	30
Accuracy(%)	98	96	97

Figure 7: Performance comparison between noise-Aware Decoder-Only Training (ADOT) and One-stage End-to-end Training (OET). PSNR represents the similarity between I_{co} and I_{en} , and robustness is the specified decoder accuracy under the noise of Cropout($p = 9\%$). It proves that ADOT guarantees image quality with comparable bit accuracy.

一旦引入噪声，OET中编码图像的质量就会明显下降，编码器以牺牲图像质量为代价加强了水印的存在。

ADOT的鲁棒性证明了编码图像可以容纳冗余的水印信息，只要充分挖掘解码器的潜力，水印仍然可以在损坏的编码图像中检索。

- 收敛时间的优势

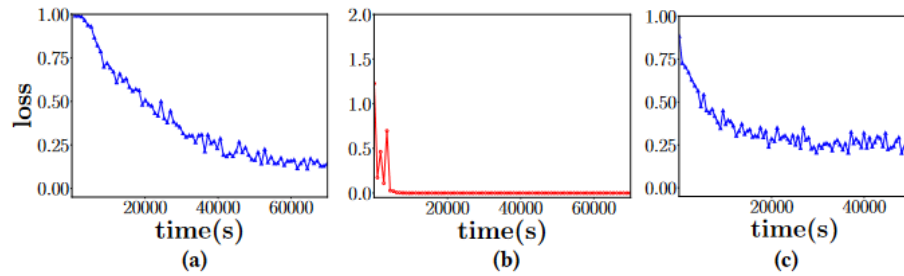


Figure 8: Loss curve comparison between OET and ADOT. (a) \mathcal{L}_1 loss curve in OET; (b) \mathcal{L}_2 loss curve in OET; (c) loss curve in ADOT. Comparing (a) with (c), loss in ADOT converges at a faster speed. In addition, (b) accounts for the unbearable encoded images in OET.

ADOT的损失以更快的速度收敛。这是因为，固定编码器后，ADOT 可以享受简化的模型并消耗更少的计算资源。

与最先进技术的比较

比较对象：HiDDeN [42]、ReDMark [1]

用组合模型来比较

Table 2: Robustness comparison with the state-of-the-art. Red represents the top accuracy value and blue takes the second place. We perform a fair robustness comparison under the condition of 33.5 PSNR. It shows that our model performs the best under most noises.

Noise Type	HiDDeN	ReDMark	Our ($S = 1$)
JPEG ($Q = 50$)	63	74.6	76.2
Cropout ($p = 30\%$)	94	92.5	97.3
Dropout ($p = 30\%$)	93	92	97.4
Crop ($p = 3.5\%$)	88	100	89
Gaussian Filter ($\sigma = 2$)	96	50	98.6

Discussion: How Does Our Model Embed Watermarks Into Images?

提出的水印过程可以表示为 $I_{en} = I_{co} + I_m$

由于 I_m 包含所有 30 位水印信息，因此很难在单个 I_m 中找到水印嵌入机制

实验尝试

1. 将全零消息 M_0 嵌入到封面图像中，并生成 I_{m0} ，作为基准来排除神经网络本身对封面图像的影响
能作为基准的原因：消息 M 的形式为 $\{-1, 1\}^L$
2. 将 M_0 位位置 p 处的位信息更改为 b ，生成 $I_{(p,b)}$ ， $b \in \{-1, 1\}$
3. 使用 *differential map* $I_{D(p,b)} = 20|I_{(p,b)} - I_{(m0)}|$ 来反映嵌入水印信息的位信息和位位置对封面图像的影响

本实验中只关心哪些像素被修改

同一位位置不同位信息下的比较

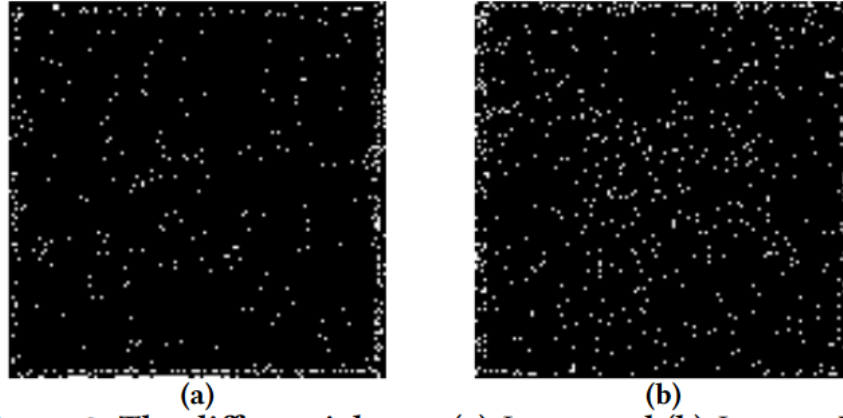


Figure 9: The differential map (a) $I_{D(0,1)}$ and (b) $I_{D(0,-1)}$. We can observe which pixels have been further modified compared I_{m0} through the white pixels. $I_{D(0,1)}$ and $I_{D(0,-1)}$ show almost different embedded pattern.

- 不同位位置同一位信息下的比较 (a) 与 (b) 比
- 通过 (a) (b) (c) 比较可以看出不同位位置之间的复杂相互作用

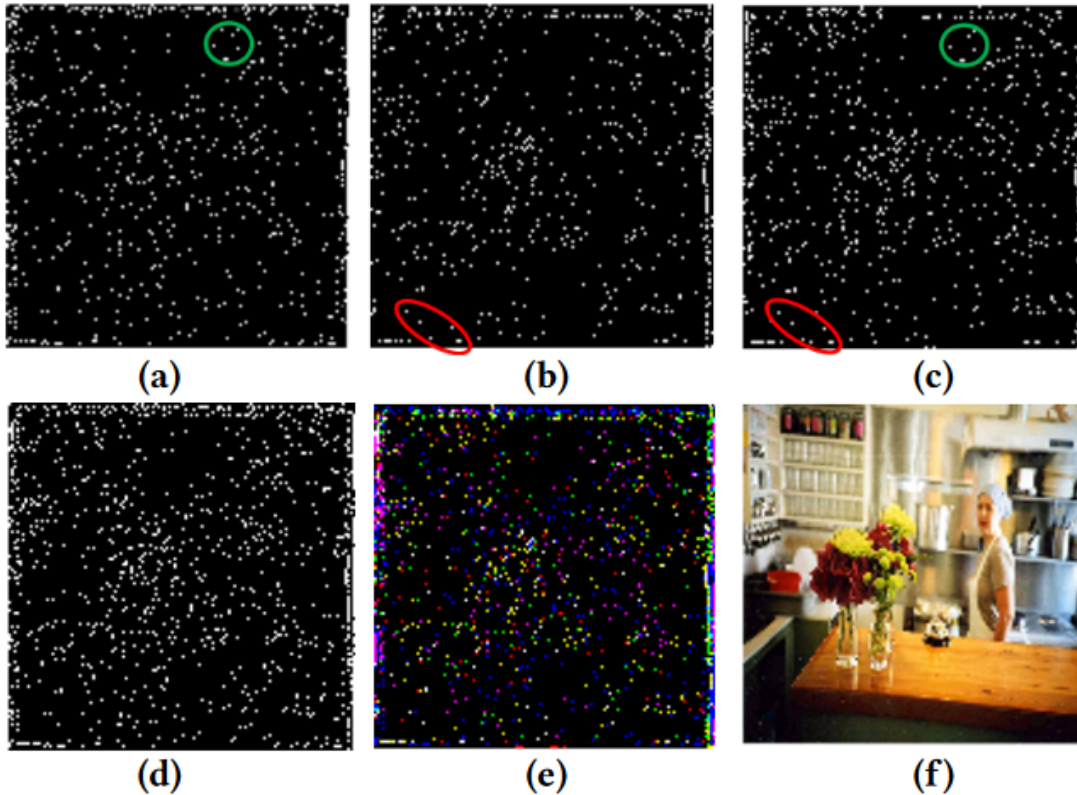


Figure 10: The comparisons of differential maps. (a) $I_{D(0,-1)}$; (b) $I_{D(1,-1)}$; (c) $I_{D(0,-1)(1,-1)}$. the Green and red oval circles show the examples of the pixels modified in the same position of the images in $I_{D(0,-1)}$, $I_{D(0,-1)(1,-1)}$ and $I_{D(1,-1)}$, $I_{D(0,-1)(1,-1)}$ respectively. (d) $I_{D(0,-1)} + I_{D(1,-1)}$; (e) Channel-merge map, where $I_{D(0,-1)}$ is blue channel, $I_{D(1,-1)}$ is green channel and $I_{D(0,-1)(1,-1)}$ is red channel. The purple and yellow pixels represent white pixels appear at the same position both in $I_{D(0,-1)}$, $I_{D(0,-1)(1,-1)}$ and $I_{D(1,-1)}$, $I_{D(0,-1)(1,-1)}$ respectively. (f) I_{en} .

结论

1. 同一位置的不同比特信息具有不同的嵌入模式
2. 不同位置的同一比特信息具有不同的嵌入模式
3. 每个位位置之间有一点相互作用，独立性有限
4. **多像素修改掩码（意思是说改变一个bit的信息也会改变多个像素的意思吗）** 表示每一点水印信息都以冗余的方式嵌入到图像中

Conclusion

介绍了一种两阶段的实用盲水印深度学习（TSDL）框架

- 第一阶段为无噪声端到端训练
- 第二阶段为针对各种噪声的噪声感知仅解码器训练，在第二阶段通过只训练解码器来引入针对黑盒噪声等不能模拟的噪声攻击的训练

该框架不仅对常见的传统高强度噪声具有鲁棒性，而且对一些黑盒噪声具有鲁棒性

与最先进的方法相比，该框架在大多数类型的噪声中都实现了最佳性能