基于可逆神经网络的大容量图像隐写技术（中文翻译版）

Shao-Ping Lu[1*] Rong Wang[1*] Tao Zhong[1] Paul L. Rosin[2]
[1]TKLNDST, CS, Nankai University, Tianjin, China
[2]School of Computer Science & Informatics, Cardiff University, UK
slu@nankai.edu.cn; nkwangrong@163.com; zei.t@qq.com; RosinPL@cardiff.ac.uk

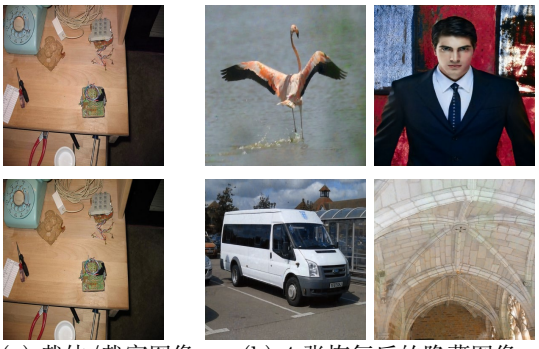
Abstract

在图像中隐藏信息一直以来都受到了广泛关注，但是如何在保证包含隐藏信息的载密图像不被检测到的情况下增加隐写容量，仍然是一个具有挑战性的问题。在本文中，我们为图像隐藏任务提出了一种大容量的可逆隐写网络（Invertible Steganography Network，ISN）。我们把图像隐写和恢复看做是图像域变换的一对可逆问题，并引入可以双向映射的可逆神经网络来解决该问题。我们利用同一模型的前向和反向映射来分别解决图像的隐藏和恢复问题。由于在两个过程中共享所有参数，我们能够同时获得高质量的载密图像和恢复的隐藏图像。另外，在我们的结构中，通过增加隐藏图像分支的通道数量就可以很自然地扩大隐写的容量。大量实验表明，随着隐写有效负载容量的显着提高，我们的 ISN 方法在视觉比较和定量分析上均达到最优。

1. 引言

隐写术是通过将一些机密数据嵌入到非机密的载体中来隐藏某些秘密数据的技术。不同于隐藏数据含义（或使其难以理解）的密码学，隐写的目的是隐藏数据的存在 [11, 42]。相应地，图像隐写就是在图像文件中隐藏数据的过程。被选来用于隐藏数据的图像称为载体图像，通过隐写术生成的图像称为载密图像，从载密图像中恢复出来的秘密图像成为重构图像。如今，图像隐写术已用于数字通信，版权保护，信息认证，电子商务和许多其他实际领域 [11]。

一个好的图像隐写系统需要同时满足隐蔽性和有效负载容量两个方面的要求 [33]。首先，隐蔽性要求载密图像应该避免引起怀疑，也就是说不应被隐写分析工具检测到隐藏数据的存在。如图 1所示，将隐藏图像嵌入到载体图像中，如果生成的载密图像在颜色和其他特征方面看起来都与载体图像相似，那么图像隐写分析技术 [18, 24] 将很难区分载体和载密图像。因此，图像隐写术本质上要求一种强大的图像表示机制，该机制可以有效地将带有隐藏图像 “噪声” 的载密图像表示为与载体图像十分相近的形式。另外，还要求这个过程是可逆的，因为在图像隐写术的恢复过程中应该从载体图像中很好地重构出隐藏图像。除此之外，为了使图像隐写术应用在实践中更加有效，另一个重要方面是将尽可能多的隐藏数据嵌入到载体图像中。



(a) 载体/载密图像 (b) 4 张恢复后的隐藏图像

图 1. (a) 我们通过将其他 4 个图像隐藏到载体图像中来生成载密图像。猜猜左列中哪个是载密图像？答：左上角和左下角分别是载密图像和原始载体图像。(b)：从载密图像中恢复出 4 个隐藏图像。这 6 张图像具有相同的分辨率。

现有的图像隐写术解决方案 [8, 40, 62] 无法同时实现大容量和高隐蔽性的隐写。传统方法通常将信息隐藏在空间域，变换域或某些自适应域中 [33]，其负载能力

*共同一作.

约为 0.2 ∼ 4 bpp (bits per pixel)。最常用的方法是将隐藏数据嵌入到最低有效位 (LSB，the least significance bits) [8] 或使用浅层视觉描述符检测出的不敏感区域，这意味着只能嵌入少量隐藏信息。最近的几种基于深度学习的隐藏方法 [4,5] 找到了一种增加隐写容量的可行方法。但因为这种图像隐写系统针对图像预处理，图像隐写和图像恢复分别设计了不同神经网络，且在整个系统中各个组件是彼此独立不共享参数的，所以很难既生成具有良好的隐蔽性的载密图像，又保证能够从中高质量地恢复出隐藏信息。

在本文中，我们提出了一种基于可逆神经网络 (INN，Invertible Neural Network) 的大容量图像隐写方法 [14,15,58]。我们将图像隐写视为一种特殊的图像域转换任务，它要求载密图像应尽可能与载体图像相似。在其逆过程中，还应该从载密图像中很好地重建隐藏图像。因此，我们将图像隐写和恢复当作一对可逆问题，从而引入可逆隐写网络 (ISN，Invertible Steganography Network) 来有效解决该问题。我们提出的新解决方案仅仅使用一个可以双向映射的 ISN 网络，其前向隐写和反向恢复过程共享所有可学习的参数。该方案使我们不仅能够有效地生成载密图像，还能高质量地恢复出隐藏图像。我们的 ISN 网络由载体和隐藏两个分支组成，分别与输入的载体图像和隐藏图像相对应，可以通过增加隐藏分支的通道数来显著提高隐写容量。大量实验表明，我们的方法在大容量条件下能够生成令人满意的载密图像，并且在相同的框架下，成功地恢复出多个隐藏图像 (图 1展示了隐藏 4 张图像的结果)。

综上所述，本文的主要贡献是：

- 我们引入 ISN 以有效解决图像隐写和恢复问题，该双射变换模型使用单个网络来有效地隐藏和恢复图像。
- 我们的方法很容易扩展为隐藏多张图像，并将隐写容量显著提高到 24 ∼ 120 *bpp*。
- 大量定性和定量实验表明我们方法在图像隐写和恢复中的结果都达到最优。

2. 相关工作

图像隐写已在学术界进行了广泛的研究 [9,33]. 在这里，我们简要讨论图像隐写术的一些代表性工作以及与可逆神经网络最相关的一些技术。

传统图像隐写算法。图像隐写术技术可以简要地分为三种类型：基于空间域的 [8,31,36–38,43,52,56]、基于变换域的 [19,26,41,42,45] 和自适应的隐写算法 [22,23,27–29,35,40]。最常用的空域隐写算法是 LSB [8]，即通过修改载体图像的最低有效位来嵌入信息。但 LSB 会在载密图像的统计信息中留下了痕迹，可以通过某些隐写分析方法轻松检测出来 [18,24,61]。其他空域隐写算法还包括像素值差法（PVD）[38,56], 直方图位移法 [43,52], 多位平面 [31,36], 调色板 [31,37] 等等。变换域隐写算法则在各种变换域中进行图像隐藏 [9,33]。例如, JSteg [42] 将隐藏信息嵌入载体图像离散余弦变换域 (DCT) 的最低有效位中。一般来说，DCT 的信息隐藏技术 [19,26,41,45] 都有比较低的隐写负载容量。

自适应的隐写算法通常采用一种通用的数据嵌入框架，将隐写分解为嵌入失真最小化和数据编码两个问题。在这类方法中，[40] 提出了一个著名框架，利用像素邻接矩阵特征 [39] 和综合征格码 [17] 进行自适应隐写。同样地，一些其他的自适应方法 [22,23,27–29,35] 设计了不同的代价函数。这些方法具有很好的隐蔽性，但在隐写容量方面仍然有普遍限制。

基于深度学习的图像隐写算法。近年来，相继提出了多种基于深度学习的图像隐写算法。这些方法可分为四大类 [10]: 基于合成的方法 [46,53], 基于生成修改概率图的方法 [50,59], 基于对抗性嵌入的方法 [49] 和基于三阶段的方法 [5,25,60,62]。

在基于合成的方法中, [46] 和 [53] 都使用生成性对抗网络（GAN）来生成更合适的载体图像。与传统的隐写方法相比，这些方法在隐写容量方面并没有明显的提升。在基于生成修改概率图的方法中，大多数方法都致力于设计满足最小失真嵌入的各种代价函数 [40]。[50] 提出了一种基于 GAN 的失真模拟框架，[59] 使用 U-Net 结构的生成器将输入图像转换为载密图像。在基于对抗嵌入的方法中，[49] 提出了一种在失真最小化框架下的对抗嵌入方案。在基于三阶段的方法中，HiDDeN [62] 和 SteganoGAN [60] 采用编码-解码结构进行信息嵌入和提取。为了抵抗隐写分析，他们引入第三个网络来扮演对抗者的角色。

最近，一种名为 DeepSteg 的方法 [4,5] 成功实现在载体图像中嵌入一张具有相同尺寸的秘密图像。该方法使用的是一个包括预处理、隐写和恢复三部分的

全卷积网络，虽然能够进行端到端的训练，但三个部分是相互独立的网络模块。与之对比，我们的方法只需训练一个可逆神经网络，隐藏和恢复过程共享所有的参数。

隐写相关应用。许多基于隐写术的应用也已经被提出。例如, Chen et al. [12] 将图像隐写技术集成到风格转换中。Wengrowski et al. [55] 引入光场信息 LFM，利用隐藏、恢复和失真模拟网络进行信息传输。Tancik et al. [48] 提出了一种称为 StegaStamp 的隐写系统。除了可感知的图像内容之外，该系统还可用于隐藏其他额外提供的信息。除此之外，还有一些有趣的工作 [51] 聚焦于通过把某些物体或者纹理变得与目标图像相似来隐藏它们。

可逆神经网络（INN）。近年来，可逆神经网络作为一种有效的图像可逆变换方法，引起了人们的广泛关注。INN 学习数据分布之间的稳定可逆映射 p_X 和潜在分布 p_Z. 不同于 CycleGAN [63] 通过构造循环损失函数训练两个生成器来实现双向映射，INN 将前向和反向传播操作包含在同一个网络模型中，从而同时实现了图像的特征编码和生成。

基于 INN 映射的开创性研究可以在之前的研究工作 NICE [14] 和 RealNVP [15] 中看到。 [20] 对这种可逆性作了进一步的解释。INN 在估计逆问题的后验概率方面也有一定优势 [2]。 [47] 通过一些合成规则利用带掩蔽的卷积构造了更灵活的 INN。 [13] 还介绍了一种基于无偏流的生成模型 [13]。此外 FFJORD [21], Glow [34], i-RevNet [32] 和 i-ResNet [6] 进一步改进了可逆耦合层来实现稠密预测并得到了不错的生成效果。由于具有强大的网络表示能力，INN 还被用于各种推理任务，如图像着色 [3], 图像缩放 [58], 图像压缩 [54] 以及视频超分 [64]。我们充分利用 INN 的双射结构和可逆性来解决隐写问题。

3. 模型方案

3.1. 总览

如图 2 (b) 所示，我们的图像隐写框架旨在高效地将多张隐藏图像嵌入载体图像并从载密图像中高质量地提取恢复出所有隐藏图像。在本文中，我们分别把载体图像和隐藏图像定义为 x_{ho} 和 x_{hi}，对应的载密图像为 y_{co}。如前文所述，我们把隐藏图像的嵌入和提取视为一对可逆问题，并把这个过程公式化为：

$$\begin{aligned} y_{co} &= f(x_{hi}, x_{ho}), \\ (\hat{x}_{ho}, \hat{x}_{hi}) &= f^{-1}(y_{co}), \end{aligned} \tag{1}$$

其中，$\hat{x}_{ho}, \hat{x}_{hi}$ 分别表示从载密图像中恢复出来的载体图像和隐藏图像。因此，我们应寻找合适的优化方法以使 y_{co} 和 $\hat{x}_{hi}$ 尽可能分别与 x_{ho} 和 x_{hi} 相似。

我们引入了单个可逆隐写网络 ISN 来同时进行特征转换，图像隐写和恢复。如式 (1)，我们利用网络的前向映射来拟合隐写函数 $f(\cdot)$, 利用其反向映射来拟合图像恢复函数 $f^{-1}(\cdot)$。其中前向映射以载体图像 x_{ho} 和隐藏图像 x_{hi} 作为输入来得到载密图像 y_{co}。在反向映射时，以载密图像 y_{co} 作为输入来恢复 $\hat{x}_{hi}$。因为前向隐写和反向恢复两个过程共享所可学习的参数，我们仅使用一个网络就能够处理这两个任务。

3.2. 可逆隐写网络 (ISN)

受到最新基于 INN（可逆隐写网络）工作 [14,15,58] 的启发，我们提出 ISN（可逆隐写网络），利用同一个网络来有效解决图像的隐写和恢复问题。如图 2 (b), 我们的 ISN 由一些可逆块堆叠而成。INN 最基本的可逆耦合层是 NICE [14] 提出的加性仿射变换。在这个模型中，对于第 l 个可逆块, 我们在通道维度将输入张量 b^l 分为 b_1^l 和 b_2^l 两部分，对应的输出分别为 b_1^{l+1} 和 b_2^{l+1}。在网络正向传播时，

$$\begin{aligned} b_1^{l+1} &= b_1^l + \phi(b_2^l), \\ b_2^{l+1} &= b_2^l + \eta(b_1^{l+1}), \end{aligned} \tag{2}$$

其中 $\phi(\cdot)$ 和 $\eta(\cdot)$ 可以是包括神经网络在内的任意函数。对于反向传播，给定 $[b_1^{l+1}, b_2^{l+1}]$, 也能很容易计算出 $[b_1^l, b_2^l]$:

$$\begin{aligned} b_2^l &= b_2^{l+1} - \eta(b_1^{l+1}), \\ b_1^l &= b_1^{l+1} - \phi(b_2^l). \end{aligned} \tag{3}$$

对于我们以图藏图的隐写算法，正向传播将隐藏图像 x_{hi} 嵌入到载体图像 x_{ho} 中。ISN 的输入包含载体和隐藏两个部分，这两个部分自然地与 b_1^l 和 b_2^l 的划分相匹配。人们经常使用仿射耦合层 [15] 来提高网络的表示能力。按照 [58], 我们对载体图像分支 b_1^l 使用

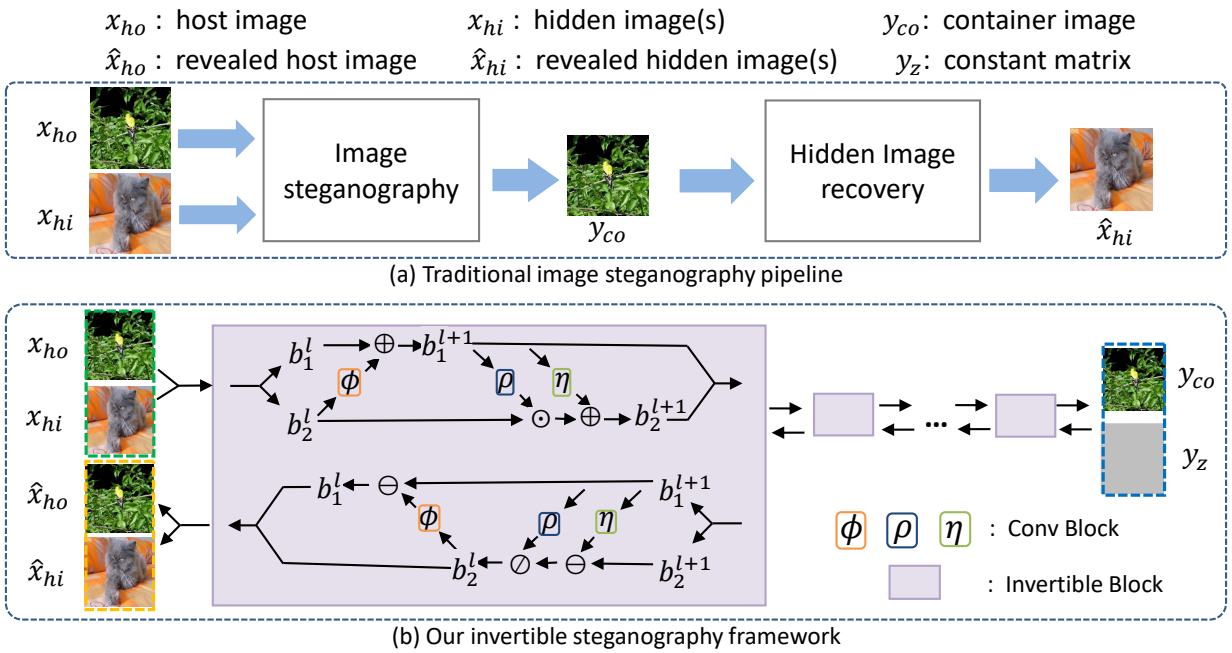


图 2. 网络结构。与传统方法（a）分别处理图像的隐写和恢复过程不同，我们引入了可逆隐写网络 ISN（b）。多个隐藏图像与载体图像拼接在一起，用作可逆网络的前向输入，并利用几个结构相同的可逆块生成载密图像。反之，反向传播有效地从载密图像中恢复出高质量的隐藏图像。

加性变换，对隐藏的图像分支 b_2^l 采用增强的仿射变换。如此以来，我们调整了网络前向映射，对应的式 (2) 改写为:

$$\begin{aligned} b_1^{l+1} &= b_1^l + \phi(b_2^l), \\ b_2^{l+1} &= b_2^l \odot \exp(\rho(b_1^{l+1})) + \eta(b_1^{l+1}), \end{aligned} \tag{4}$$

其中 $\exp(\cdot)$ 和 $\rho(\cdot)$ 分别是指数函数和任意函数。$\odot$ 是 Hadamard 算子. 这是增强版可逆块的一个变体。相应地反向传播表示为:

$$\begin{aligned} b_2^l &= (b_2^{l+1} - \eta(b_1^{l+1})) \odot \exp(-\rho(b_1^{l+1})), \\ b_1^l &= b_1^{l+1} - \phi(b_2^l). \end{aligned} \tag{5}$$

可逆块结构如图 2 (b) 所示。注意对 $\rho(\cdot)$ 的 $\exp(\cdot)$ 函数在图中进行了省略。

我们的 ISN 在生成载密图像 y_{co} 时，需要引入一个常数矩阵 y_z (见 图 2 (b) 的最右侧)。当我们要将一个 RGB 图像隐藏到另一个 RGB 图像中时，每个可逆块的输入和输出都有 6 个通道，这意味着正向输出也需要有 6 个通道。然而，我们只需 3 个通道就能表示 y_{co}. 为了保持可逆网络两侧的通道数和特征信息的一致性，我们将除 y_{co} 外剩余的 3 个信道设置为常数矩阵 y_z.

值得注意的是，我们的 ISN 可以灵活地嵌入多个隐藏图像。为了实现这一点，我们直接在通道维度拼接 n 个隐藏图像得到一个 $3n$ 通道的 x_{hi}，同时增加在隐藏分支 b_2 的特征通道数量即可，不需要改变网络的整体结构。

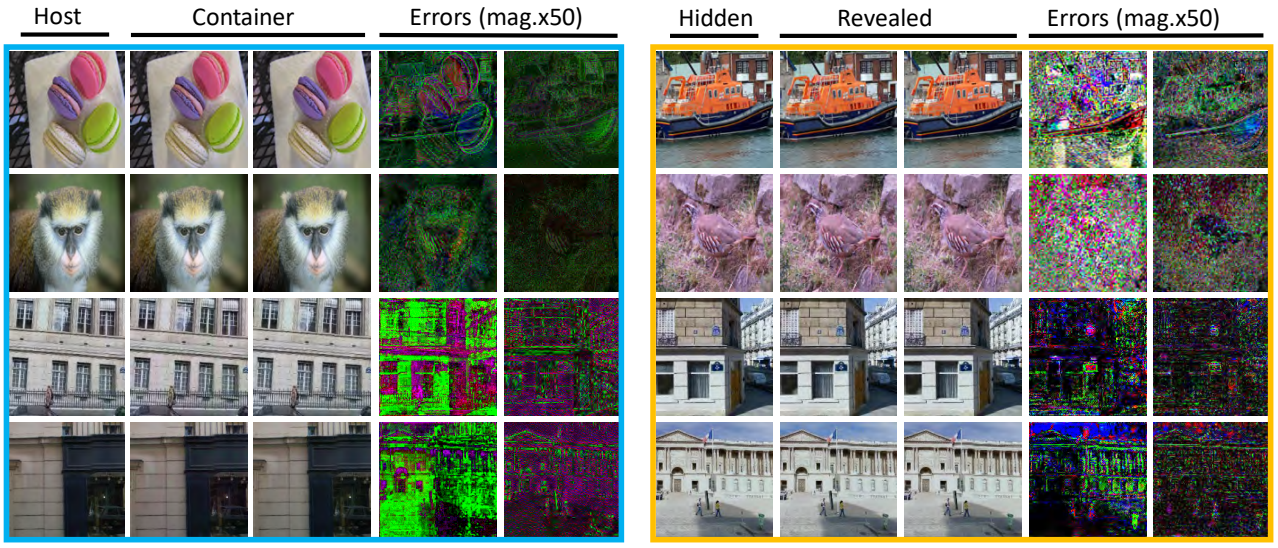
3.3. 损失函数

我们的目标是使载密图像 y_{co} 和被恢复的隐藏图像 $\hat{x}_{hi}$ 尽可能地与载体图像 x_{ho} 和隐藏图像 x_{hi} 一致。因此，我们为 y_{co} 和 $\hat{x}_{hi}$ 引入以下两种损失函数:

$$\begin{aligned} \mathcal{L}_{co} &= \mathcal{F}(y_{co}, x_{ho}), \\ \mathcal{L}_{hi} &= \mathcal{F}(\hat{x}_{hi}, x_{hi}). \end{aligned} \tag{6}$$

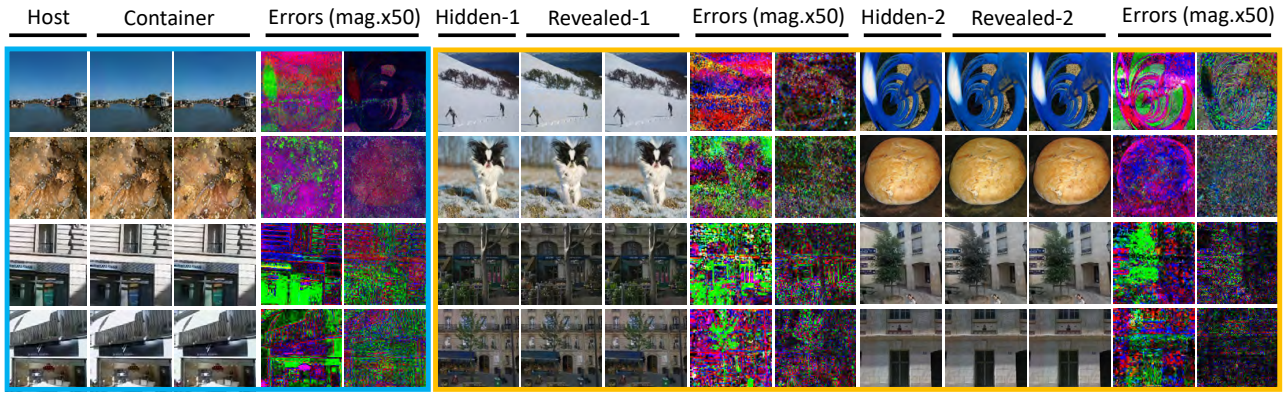
这里 $\mathcal{F}$ 是像素级距离函数。此外，对于网络的另外两个输出结果，恢复的载体图像 x_{ho} 和常数矩阵 y_z，也分别构造了以下两个损失函数:

$$\begin{aligned} \mathcal{L}_{ho} &= \mathcal{F}(\hat{x}_{ho}, x_{ho}), \\ \mathcal{L}_{z} &= \mathcal{F}(\hat{y}_{z}, y_{z}). \end{aligned} \tag{7}$$



(a) 原图 (b) [5] (c) ISN (d) [5] (e) ISN (f) 原图 (g) [5] (h) ISN (i) [5] (j) ISN

图 3. 隐藏和恢复一张图像的视觉比较。



(a) 原图 (b) [5] (c) ISN (d) [5] (e) ISN (f) 原图 (g) [5] (h) ISN (i) [5] (j) ISN (k) 原图 (l) [5] (m) ISN (n) [5] (o) ISN

图 4. 隐藏和恢复两张图像的视觉比较。

这两个损失进一步约束系统优化到最优解，以生成所需的图像。与 [58] 相同，我们分别对正向和反向的输出使用 l_2 和 l_1 损失函数。综上述，我们的最终损失函数为：

$$\mathcal{L} = \alpha_{co}\mathcal{L}_{co} + \alpha_z\mathcal{L}_z + \alpha_{ho}\mathcal{L}_{ho} + \alpha_{hi}\mathcal{L}_{hi}, \qquad (8)$$

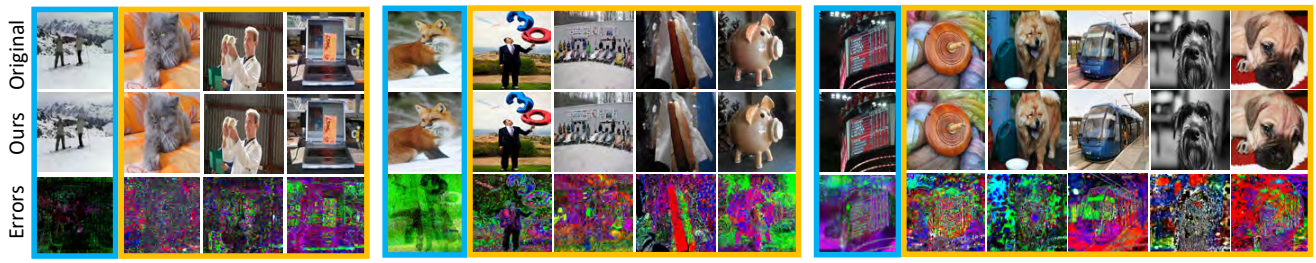
其中 $\alpha_{co}, \alpha_z, \alpha_{ho}, \alpha_{hi}$ 是上述相应损失的权重。

4. 实验结果

4.1. 实现细节

我们的 ISN 是通过 PyTorch 实现的，并使用 Nvidia Titan 2080Ti 显卡进行加速。我们使用 AdaMax 优化器来训练我们的网络，其中 $\beta_1 = 0.9$, $\beta_2 = 0.999$, 学习率为 0.0002, batch size 为 2。我们的网络包含多个可逆模块，每个可逆模块都使用三个 5 层的 DenseNet 子模块来分别作为 ϕ, ρ 和 η 函数。可逆块的数量和损失函数的权重与隐写容量即隐藏图像的数量有关（详细信息参见第 4.4小节）。

我们在包含大量自然和人为景观的 ImageNet [44] 和 Paris StreetView [16] 数据集上对我们的模型进行训练和测试。我们从 ImageNet 中随机抽取 100000 张图像训练和 1000 张图像测试。对于 Paris StreetView 数据集，我们有 14900 张训练图像和 100 张测试图像。我们随机剪裁 144×144 大小的图像块来训练，同时采用随机翻转和旋转进行数据增强。在训练期间，我们交替进行网络的正向和反向传播。对于每次迭代，我们



(a) 隐藏 3 张图片. (b) 隐藏 4 张图片. (c) 隐藏 5 张图片.

图 5. 隐藏多个图像的结果，子图 (a), (b) 和 (c) 分别代表隐藏 3 ~5 张图片的结果。带有蓝色边框的是载体图像，带有橙色边框的是隐藏图像。在每个子图中，第一行是原始图像，中间行是我们生成的结果，而第三行是把他们之间的误差放大 ×50 倍的结果。

表 1. PSNR/SSIM 的客观比较。-$h1$ 和 -$h2$ 分别表示隐藏 1 张和 2 张图像，(c) 意味着跨域测试，即在一个数据集上训练的模型直接在另一个数据集上进行测试而没有微调。

方法	ImageNet		Paris StreetView	
	载密图像	恢复图像	载密图像	恢复图像
ISN-$h1$	38.05/.954	35.38/.955	40.49/.980	43.33/.991
ISN-$h1$ (c)	36.48/.940	34.92/.950	39.28/.977	40.41/.985
[5]-$h1$	36.02/.946	32.75/.933	36.80/.986	39.03/.984
[5]-$h1$ (c)	30.12/.938	29.53/.897	38.29/.975	35.86/.971
ISN-$h2$	36.86/.945	32.21/.920	39.14/.971	39.05/.982
ISN-$h2$ (c)	35.57/.932	32.04/.926	38.69/.969	35.12/.962
[5]-$h2$	30.18/.919	29.17/.898	37.14/.978	34.73/.964
[5]-$h2$ (c)	29.85/.931	25.19/.833	35.20/.963	33.23/.955

的网络首先执行前向计算 $F(x_{ho}, x_{hi})$ 来获得 $(y_{co}, \hat{y}_z)$，接着执行反向计算 $F^{-1}(y_{co}, y_z)$，最后计算相应的 4 个损失函数并更新网络参数。

用于隐藏一张图像的 ISN 大约需要训练一天，进行 500000 次迭代。在进行推理时，隐藏并恢复一张 380×380 大小的图像大概需要 0.07 秒。我们还在 MindSpore [1] 和其他平台实现了我们的模型。具体来说，我们模型的推理速度在 Jittor 深度学习框架 [30] 上提高了 12% 。

4.2. 对比实验

在这里，我们进行了一些对比试验，特别是与最新提出的 [5] 方法。没有与其他一些基于 CNN 的方法，比如 HiDDeN [62] 和 SteganoGAN [60] 进行比较，是因为它们能达到的有效载荷能力 (<4.5 bpp) 还只是与传统算法相当。我们在 PyTorch 上复现了 [5] 的模型，并在同样的 ImageNet 和 Paris StreetView 数据集上对其进行训练和测试。我们用 PSNR（峰值信噪比）和 SSIM（结构相似性）度量标准对图像质量进行客观评估。可以看到到我们复现模型的客观评价指标 (见表 1) 略低于论文 [5] 中的值。这是由随机选择的测试数据不同所致。当隐藏两个图像时，我们使用其平均 PSNR 来衡量恢复图像的重建质量。表 1 的结果表明我们的方法在隐藏一张和两张图像时都取得了更好的效果。有趣的是，表 1 还表明，我们的模型即便只在数据量少的 Pairs StreetView 数据集上训练，并直接在 ImageNet 上测试的结果仍然是可以接受的。

我们和 [5] 的视觉比较见 图 3。由于篇幅所限，这里我们只为每个数据集展示了两个示例（更多示例在补充材料中）。为了更好地说明原始图像和生成图像之间的差异，我们将它们之间的逐像素误差放大 50 倍后进行显示。可以观察到，我们生成的载密图像和恢复的隐藏图像的误差都小于 [5], 这与客观比较是一致的。总体而言，这些实验表明，我们的 ISN 隐藏一张或者两张图像时，在定量和定性方面都能获得最佳结果。

4.3. 隐藏多张图像

在这里，我们通过嵌入多张图像来探索 ISN 的最大隐写容量。首先，如图 4我们将两幅图像嵌入到载体中，并和原始图像进行视觉比较。更近一步，图 5 将隐藏 3~5 图像的结果可视化，其中带有蓝色边框的是载体图像，带有橙色边框的是隐藏图像。显然，在如此高的隐写容量下，我们的 ISN 仍然可以获得令人满意的载密图像，并且可以高质量地恢复所有的隐藏图像。

在图 6中，我们进一步计算了隐藏不同数量图像时，载体图像和恢复的隐藏图像的平均 PSNR。在每一类实验中，我们随机选取 100 张图片进行测试。同样，隐藏图像的 PSNR 对应于每个载密图像所隐藏的多张图像的平均值。如 图 6所示，载体图像的平均 PSNR 值随着隐藏图像数量的增加而降低。容易理解，隐藏和恢复多个隐藏图像的信息会变得越来越困难。尽管如此，即使对于 5 个隐藏图像的极端情况，在载密图像具有

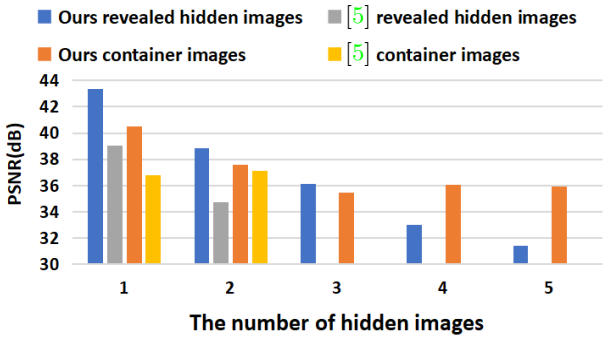


图 6. 隐藏和恢复 1~5 张图像的平均 PSNR 。

良好的视觉感知能力 (~36 dBs) 的同时，恢复的隐藏图像的 PSNR 仍高于 31dB。

4.4. 消融实验

表 2. 隐藏 1~ 2 图像的消融实验。

α_{co}	1 张隐藏图像		2 张隐藏图像	
	载密图像	恢复图像	载密图像	恢复图像
2	27.64/.908	41.38/.994	27.08/.856	38.04/.981
4	29.10/.935	42.26/.994	28.84/.894	38.15/.986
8	33.30/.961	41.16/.990	29.86/.922	37.48/.983
16	35.64/.974	41.99/.990	35.52/.932	39.26/.984
32	40.49/.980	43.33/.991	37.60/.958	38.87/.982
64	42.40/.986	40.73/.988	39.14/.971	39.05/.982

表 3. 隐藏 4 张图像的消融实验。

α_{co}	8 个 InvBlocks		16 个 InvBlocks	
	载密图像	恢复图像	载密图像	恢复图像
4	27.58/.779	32.90/.945	26.97/.787	34.66/.960
32	33.63/.928	31.61/.934	34.58/.923	33.22/.949
64	36.53/.957	31.12/.928	36.03/.955	33.02/.942

消融实验是在 Paris StreetView 数据集上进行的。在这里，我们主要讨论对最终结果影响最大的载密图像的损失权重和可逆块的数量。有关子模块选择和损失函数调整的更多实验，请参阅补充资料。

如表 2所示，当嵌入一到两张图像时，我们的 ISN 很容易恢复出高质量的隐藏图像。通过简单地调整损失函数式 (8) 中载密图像的权重 α_{co}, ISN 就可以获得理想的载密图像。在隐藏 2 个图像时，在不降低载密图像质量（仍高于 38 dB）的情况下，将 α_{co} 从 2 调整到 64，就使得载密图像的 PSNR 和 SSIM 分别提升 +12.06 dBs 和 +0.115。

同样，当隐藏 4 个图像时，增大 α_{co} 也能显著提高载密图像的质量 (见表 3)。然而，却很难通过调整权重来提高隐藏图像的恢复质量。在此表中，如果我们仅使用具有 8 个可逆块的 ISN，则所恢复的 4 个隐藏图像的平均 PSNR 会始终小于 33 dB。通过将可逆块的数量从 8 增加到 16，恢复的隐藏图像将获得 +1.9 dB 的提升，同时载密图像也高于 36 dB(在 表 3最后一行)。换句话说，当处理更多隐藏图像时，可以通过适当增加可逆块的数量来提高 ISN 法的隐写和恢复能力。根据表 2 和表 3, 藏一张图像时，我们设 α_{co} 为 32，藏多张图像时设 α_{co} 为 64 以确保载密图像与载体图像足够相似。其他损失函数的权重 $\alpha_z, \alpha_{ho}, \alpha_{hi}$ 都设置为 1。

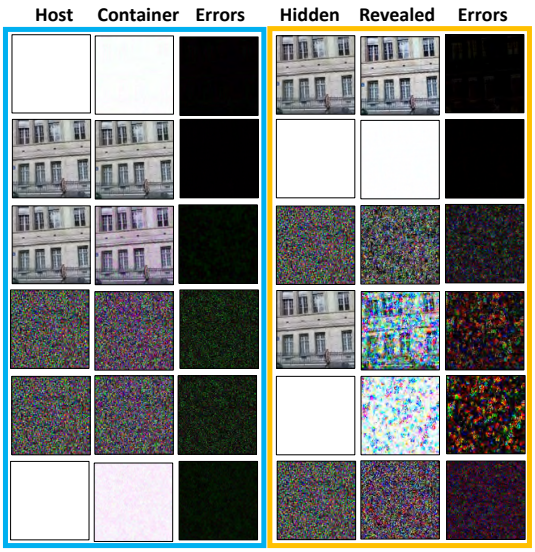


图 7. 在某些极端情况下的视觉结果，其中载体或隐藏图像是单色，自然或随机噪声图像。

5. 讨论

5.1. 极端情况

为了探索我们提出的 ISN 的隐写能力，我们对一些包括自然图像、单色图像和随机噪声图像在内的极端图像进行了实验。对于每两幅图像，我们首先选择其中一幅嵌入到另一幅图像中。之后，我们通过调换这两个图像的角色来重复上述实验。从 图 7的前两行可以观察到，将自然图像嵌入单色图像时，我们的方法效果很好，反之亦然。然而，其他结果（后四行）表明，如果将噪声图像用作隐藏图像或载体图像，则我们的方法很难准确恢复出隐藏图像。

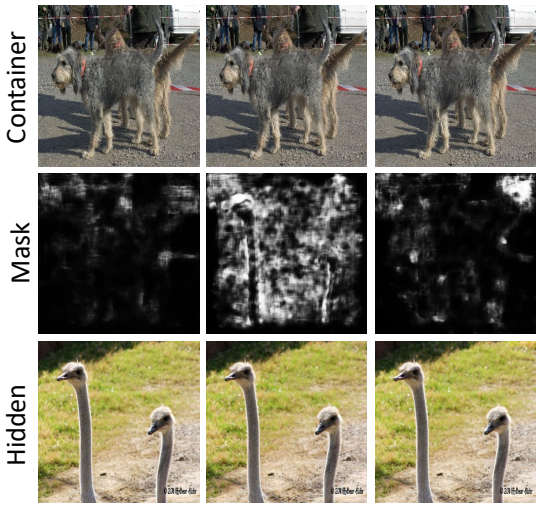


图 8. 伪造检测。第二行中的 Mask 分别是第一行中载密图像由 ManTra-Net [57] 检测的结果。第三行是隐藏图像。

5.2. 隐写分析

在这里，我们使用两个广泛使用的开源工具 [7,57] 对 ISN 生成的载密图像进行隐写分析。第一个工具 ManTra-Net [57], 可用于检测 385 种图像编辑类型。我们把该工具对我们方法和 [5] 的检测结果进行了比较。如图 8所示, 我们载密图像被 [57] 检测出的异常信息和原始的载体图像很相似。相反，使用 [5](图 8的中间) 生成的载密图像会被检测到关于隐藏图像的更多信息。这证明了我们 ISN 的有效性。

另外一个隐写分析工具 StegExpose [7]，是专为 LSB 隐写检测而设计，包括四种众所周知的隐写分析方法。图 9是 [5] 和 ISN 在 StegExpose 下检测结果的 ROC 曲线。这两条可比较的曲线表明 StegExpose 检测对 ISN 和 [5] 都不完全奏效。同样有趣的是， [5] 的检测曲线略好于我们的，这与之前 PSNR 指标报告的结果是相反的。

5.3. 加密

如前文所述，训练过程中我们将隐藏图像分支前向传播的输出约束为一个常数矩阵 y_z 。对于我们在第 4小节中的实验, 所有 y_z 参数都被设置为 0.5 的固定值, 这样可以很好地保持网络结构的一致性，并且可以按照预期将隐藏分支的信息转移到载体分支。

能更进一步将 y_z 用作隐藏图像提取的秘钥吗? 我们在训练过程中设置 y_z 为不同的纹理图案, 并尝试在

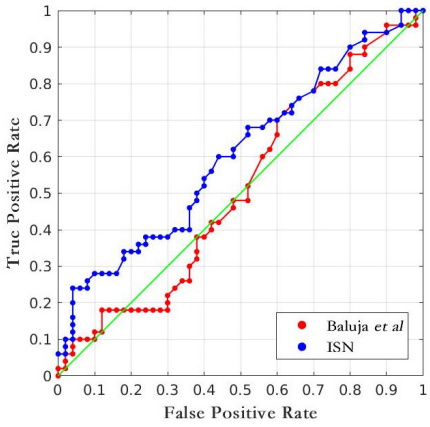


图 9. [5] 和 ISN 生成的载密图像在 StegExpose [7] 设置不同阈值下检测获得的 ROC 曲线。

图 10. 通过设置 y_z 为不同的纹理图案来提取隐藏信息。左上角和左下角分别是载密图像和隐藏图像。第一行的其余 5 张图像是 y_z 的纹理图案, 它们的底部是从载密图像中恢复的相应结果。

y_z 未知的情况下恢复隐藏图像。图 10 显示了当 y_z 使用不同纹理时，隐藏图像恢复的结果。我们看到只有 y_z 设置正确即与训练时相同, 隐藏图像才可以高质量地恢复出来。还要注意的是，虽然用不恰当的 y_z 恢复的图像是扭曲的，但隐藏图像的部分内容仍可以辨认。

6. 总结

在本文中，我们提出了一种用于图像隐写的可逆隐写网络 (ISN)，其特点是利用同一网络的前向和反向传播分别进行隐藏图像的嵌入和提取。我们的方法显著提高了隐写容量，并且在保证隐蔽性的同时可以很容易地扩展成为隐藏多张图像的版本。大量实验表明，随着隐写容量的显着提高，我们的 ISN 方法在视觉比较和定量分析上均达到最优。

致谢 感谢各位审稿人的宝贵意见。感谢国家自然科学基金 (No. 61972216) 和天津市自然基金 (No. 18JCYBJC41300 and No. 18ZXZNGX00110) 对本工作的支持。本文的通讯作者是卢少平。

参考文献

[1] MindSpore. https://www.mindspore.cn/, 2020. 6

[2] Lynton Ardizzone, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks. In ICLR, 2018. 3

[3] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. arXiv preprint arXiv:1907.02392, 2019. 3

[4] Shumeet Baluja. Hiding images in plain sight: Deep steganography. In NeurIPS, pages 2069–2079, 2017. 2

[5] Shumeet Baluja. Hiding images within images. IEEE Trans. Pattern Anal. Mach. Intell., 2019. 2, 5, 6, 7, 8

[6] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In ICML, pages 573–582, 2019. 3

[7] Benedikt Boehm. Stegexpose - a tool for detecting LSB steganography. arXiv preprint arXiv:1410.6656, 2014. 8

[8] Chi-Kwong Chan and Lee-Ming Cheng. Hiding data in images by simple LSB substitution. PR, 37(3):469–474, 2004. 1, 2

[9] Yambem Jina Chanu, Kh Manglem Singh, and Themrichon Tuithung. Image steganography and steganalysis: A survey. Int. J. Comput. Vision., 52(2), 2012. 2

[10] Marc Chaumont. Deep learning in steganography and steganalysis from 2015 to 2018. arXiv preprint arXiv:1904.01444, 2019. 2

[11] Abbas Cheddad, Joan Condell, Kevin Curran, and Paul Mc Kevitt. Digital image steganography: Survey and analysis of current methods. Signal processing, 90(3):727–752, 2010. 1

[12] Hung-Yu Chen, I-Sheng Fang, Chia-Ming Cheng, and Wei-Chen Chiu. Self-contained stylization via steganography for reverse and serial style transfer. In IJCAI, March 2020. 3

[13] Ricky TQ Chen, Jens Behrmann, David K Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling. In NeurIPS, pages 9916–9926, 2019. 3

[14] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear independent components estimation. arXiv preprint arXiv:1410.8516, 2014. 2, 3

[15] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. arXiv preprint arXiv:1605.08803, 2016. 2, 3

[16] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei Efros. What Makes Paris Look like Paris? ACM Trans. Graph., 31(4), 2012. 5

[17] Tomáš Filler, Jan Judas, and Jessica Fridrich. Minimizing embedding impact in steganography using trellis-coded quantization. In Media forensics and security II, volume 7541, page 754105, 2010. 2

[18] Jessica Fridrich, Miroslav Goljan, and Rui Du. Detecting LSB steganography in color, and gray-scale images. IEEE Trans. Multimedia, 8(4):22–28, 2001. 1, 2

[19] Jessica Fridrich, Tomáš Pevnỳ, and Jan Kodovskỳ. Statistically undetectable JPEG steganography: dead ends challenges, and opportunities. In workshop on Multimedia & security, pages 3–14, 2007. 2

[20] Anna C Gilbert, Yi Zhang, Kibok Lee, Yuting Zhang, and Honglak Lee. Towards understanding the invertibility of convolutional neural networks. In IJCAI, pages 1703–1710, 2017. 3

[21] Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. arXiv preprint arXiv:1810.01367, 2018. 3

[22] L. Guo, J. Ni, and Y. Q. Shi. An efficient JPEG steganographic scheme using uniform embedding. In WIFS, pages 169–174, 2012. 2

[23] L. Guo, J. Ni, and Y. Q. Shi. Uniform embedding for efficient JPEG steganography. IEEE Trans. Inf. Forensics Secur., 9(5):814–825, 2014. 2

[24] Tariq Al Hawi, MA Qutayri, and Hassan Barada. Steganalysis attacks on stego-images using stego-signatures and statistical image properties. In TENCON, pages 104–107, 2004. 1, 2

[25] Jamie Hayes and George Danezis. Generating steganographic images via adversarial training. In NeurIPS, pages 1954–1963, 2017. 2

[26] Stefan Hetzl and Petra Mutzel. A graph–theoretic approach to steganography. In IFIP international conference on communications and multimedia security, pages 119–128, 2005. 2

[27] Vojtěch Holub and Jessica Fridrich. Designing steganographic distortion using directional filters. In WIFS, pages 234–239, 2012. 2

[28] Vojtěch Holub and Jessica Fridrich. Digital image steganography using universal distortion. In workshop on Information hiding and multimedia security, pages 59–68, 2013. 2

[29] Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark. Universal distortion function for steganography in an arbitrary domain. EURASIP Journal on Information Security, 2014(1):1, 2014. 2

[30] Shi-Min Hu, Dun Liang, Guo-Ye Yang, Guo-Wei Yang, and Wen-Yang Zhou. Jittor: a novel deep learning framework with meta-operators and unified graph execution. Information Sciences, 63(222103):1–222103, 2020. 6

[31] Shoko Imaizumi and Kei Ozawa. Multibit embedding algorithm for steganography of palette-based images. In PSIVT, pages 99–110, 2013. 2

[32] Jörn-Henrik Jacobsen, Arnold Smeulders, and Edouard Oyallon. i-RevNet: Deep invertible networks. In ICLR, 2018. 3

[33] Inas Jawad Kadhim, Prashan Premaratne, Peter James Vial, and Brendan Halloran. Comprehensive survey of image steganography: Techniques, evaluations, and trends in future research. Neurocomputing, 335:299–326, 2019. 1, 2

[34] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In NeurIPS, pages 10215–10224, 2018. 3

[35] B. Li, M. Wang, J. Huang, and X. Li. A new cost function for spatial image steganography. In ICIP, pages 4206–4210, 2014. 2

[36] Bui Cong Nguyen, Sang Moon Yoon, and Heung-Kyu Lee. Multi bit plane image steganography. In IWDW, pages 61–70, 2006. 2

[37] Michiharu Niimi, Hideki Noda, Eiji Kawaguchi, and Richard O Eason. High capacity and secure digital steganography to palette-based images. In ICIP, volume 2, pages II–II, 2002. 2

[38] Feng Pan, Jun Li, and Xiaoyuan Yang. Image steganography method based on pvd and modulus function. In ICECC, pages 282–284, 2011. 2

[39] Tomáš Pevny, Patrick Bas, and Jessica Fridrich. Steganalysis by subtractive pixel adjacency matrix. IEEE Trans. Inf. Forensics Secur., 5(2):215–224, 2010. 2

[40] Tomáš Pevnỳ, Tomáš Filler, and Patrick Bas. Using high-dimensional image models to perform highly undetectable steganography. In International Workshop on Information Hiding, pages 161–177, 2010. 1, 2

[41] N. Provos. Defending against statistical steganalysis. In Usenix security symposium, volume 10, pages 323–336, 2001. 2

[42] N. Provos and P. Honeyman. Hide and seek: an introduction to steganography. IEEE Security Privacy, 1(3):32–44, 2003. 1, 2

[43] Chuan Qin, Chin-Chen Chang, Ying-Hsuan Huang, and Li-Ting Liao. An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism. IEEE Trans. Circuits Syst. Video Technol., 23(7):1109–1118, 2012. 2

[44] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. Int. J. Comput. Vision., 115(3):211–252, 2015. 5

[45] Phil Sallee. Model-based steganography. In IWDW, pages 154–167, 2003. 2

[46] Haichao Shi, Jing Dong, Wei Wang, Yinlong Qian, and Xiaoyu Zhang. SSGAN: secure steganography based on generative adversarial networks. In PCM, pages 534–544, 2017. 2

[47] Yang Song, Chenlin Meng, and Stefano Ermon. Mintnet: Building invertible neural networks with masked convolutions. In NeurIPS, pages 11004–11014, 2019. 3

[48] Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In CVPR, June 2020. 3

[49] Weixuan Tang, Bin Li, Shunquan Tan, Mauro Barni, and Jiwu Huang. CNN-based adversarial embedding for image steganography. IEEE Trans. Inf. Forensics Secur., 14(8):2074–2087, 2019. 2

[50] W. Tang, S. Tan, B. Li, and J. Huang. Automatic steganographic distortion learning using a generative adversarial network. IEEE Signal Processing Letters, 24(10):1547–1551, 2017. 2

[51] Qiang Tong, Song-Hai Zhang, Shi-Min Hu, and Ralph R Martin. Hidden images. In NPAR, pages 27–34, 2011. 3

[52] Piyu Tsai, Yu-Chen Hu, and Hsiu-Lien Yeh. Reversible image hiding scheme using predictive coding and histogram shifting. Signal processing, 89(6):1129–1143, 2009. 2

[53] Denis Volkhonskiy, Ivan Nazarov, and Evgeny Burnaev. Steganographic generative adversarial networks. In ICMV, volume 11433, page 114333M, 2020. 2

[54] Yaolong Wang, Mingqing Xiao, Chang Liu, Shuxin Zheng, and Tie-Yan Liu. Modeling lost information in lossy image compression. arXiv preprint arXiv:2006.11999, 2020. 3

[55] Eric Wengrowski and Kristin Dana. Light field messaging with deep photographic steganography. In CVPR, June 2019. 3

[56] Da-Chun Wu and Wen-Hsiang Tsai. A steganographic method for images by pixel-value differencing. Pattern recognition letters, 24(9-10):1613–1626, 2003. 2

[57] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. ManTra-Net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In CVPR, pages 9543–9552, 2019. 8

[58] Mingqing Xiao, Shuxin Zheng, Chang Liu, Yaolong Wang, Di He, Guolin Ke, Jiang Bian, Zhouchen Lin, and Tie-Yan Liu. Invertible image rescaling. ECCV, 2020. 2, 3, 5

[59] J. Yang, D. Ruan, J. Huang, X. Kang, and Y. Shi. An embedding cost learning framework using GAN. IEEE Trans. Inf. Forensics Secur., 15:839–851, 2020. 2

[60] Kevin Alex Zhang, Alfredo Cuesta-Infante, Lei Xu, and Kalyan Veeramachaneni. SteganoGAN: High capacity image steganography with GANs. arXiv preprint arXiv:1901.03892, 2019. 2, 6

[61] Li Zhi, Sui Ai Fen, and Yang Yi Xian. A LSB steganography detection algorithm. In PIMRC, volume 3, pages 2780–2783, 2003. 2

[62] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In ECCV, pages 657–672, 2018. 1, 2, 6

[63] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In ICCV, Oct 2017. 3

[64] Xiaobin Zhu, Zhuangzi Li, Xiao-Yu Zhang, Changsheng Li, Yaqi Liu, and Ziyu Xue. Residual invertible spatio-temporal network for video super-resolution. In AAAI, volume 33, pages 5981–5988, 2019. 3