Hiding Images in Plain Sight: Deep Steganography 于众目睽睽之下隐藏图像：深度隐写术

转载

c2a2o2 于 2019-11-18 12:27:50 发布 2420 收藏 13
分类专栏： 图像特征
原文链接：https://www.jianshu.com/p/6848888e770e

Hiding Images in Plain Sight: Deep Steganography
于众目睽睽之下隐藏图像：深度隐写术

1.摘要

隐写术是将秘密信息隐藏在另一条普通信息中的一种实践。通常，隐写术用于在较大图像的嘈杂区域中不显眼地隐藏小消息。在本研究中，作者尝试将一个全尺寸彩色图像放置在另一个相同尺寸的图像中。深层神经网络同时被训练来创建隐藏和揭示过程，并被设计成专门作为一对工作。该系统对从ImageNet数据库中随机抽取的图像进行训练，并能在各种来源的自然图像上很好地工作。除了演示深度学习在隐藏图像中的成功应用之外，还仔细研究了如何实现结果，并探索其扩展应用。与许多流行的将秘密信息编码在载体图像最低有效位的隐写方法不同，这个方法压缩并将秘密图像的表示分布到所有可用位。

2.隐写术的引言

隐写术是一种被掩盖或隐藏写作的艺术；这个术语本身可以追溯到15世纪，那时信息被物理隐藏。在现代隐写术中，目的是秘密地传递数字信息。隐写过程将一个隐藏的信息放在一个称为载体的传输媒介中。载体是可以公开可见的。为了增加安全性，还可以对隐藏的消息进行加密，从而增加感知的随机性，降低内容发现的可能性，即使检测到消息的存在。有关隐写术和隐写分析（发现隐藏信息的过程）的很好的介绍可以在[1-5]中找到。

隐写术的信息隐藏有许多广为人知的恶意应用，例如通过在公共网站上发布的图片中隐藏信息来策划和协调犯罪活动，使得通信和接收者难以发现[6]。然而，除了大量的误用之外，隐写术方法的一个常见用例是通过数字水印嵌入作者信息，而不损害内容或图像的完整性。

较好的隐写术带来的挑战是，嵌入消息可以改变载体的外观和底层统计数据。变更的数量取决于两个因素：第一，要隐藏的信息的数量。常用的方法是隐藏图像中的文本消息。隐藏的信息量以每像素比特数（bpp）为单位进行衡量。通常，信息量设置为0.4bpp或更低。信息越长，bpp越大，因此载体的改动越多[6，7]。第二，改变的数量取决于载体图像本身。将信息隐藏在噪声大、充满高频的图像区域中，会比隐藏在平坦区域中产生更少的人类可检测的扰动。估计一个载体图像可以隐藏多少信息可以在[8]中找到。

最常见的隐写术方法是操纵图像的最低有效位（least significant bits，LSB）来放置秘密信息——无论是统一还是自适应地执行，通过简单的替换或更高级的方案[9，10]。尽管图像和音频文件的统计分析通常不可见，但可以揭示结果文件是否与未经修改的文件不同。高级方法试图通过创建和匹配可能覆盖图像集的一阶和二阶统计的模型来保留图像统计；最流行的方法之一是命名为HUGO[11]。HUGO通常使用相对较小的信息（<0.5bpp）。与之前的研究相比，作者使用神经网络隐式地模拟自然图像的分布，并将更大的信息（全尺寸图像）嵌入载体图像中。

尽管最近通过将深层神经网络与隐写术结合取得了令人印象深刻的结果[12–14]，但将神经网络纳入隐藏过程本身的尝试相对较少[15–19]。其中一些研究使用深度神经网络（DNNs）来选择用文本消息的二进制表示替换图像中的哪些最低有效位。其他研究使用DNNs来确定从容器图像中提取哪些位。与之相反的是，在我们的工作中，神经网络决定了在哪里放置秘密信息以及如何有效地编码；隐藏的信息散布在图像中的各个位上。利用与编码器同时训练的译码器网络来揭示秘密图像。请注意，网络仅经过一次训练，并且独立于载体图像和秘密图像。

本文的目标是在另一个NNRGB载体图像中视觉隐藏一个完整的NNRGB像素的秘密图像，对载体图像的失真最小（每个颜色通道为8位）。然而，与以往的研究不同的是，隐藏的文本信息必须以完美的重建方式发送，作者放宽了秘密图像无损接收的要求。相反，作者希望在载体质量和秘密图像方面找到可接受的折衷办法（这将在下一节中描述）。作者还简要讨论了秘密信息存在的可发现性。以前的研究表明，隐藏消息比特率可以被发现低至0.1bpp；作者的比特率是10×-40×更高。尽管在视觉上很难发现，鉴于隐藏信息的数量巨大，我们不希望在统计分析中隐藏秘密信息的存在。尽管如此，我们仍将证明使用常用的方法找不到它，并根据需要，就如何权衡存在发现的困难与重建质量给出了有希望的指导。

架构和错误传播

虽然隐写术经常与密码学混为一谈，但在我们的方法中，最接近的模拟是通过自动编码网络进行图像压缩。经过训练的系统必须学会将秘密图像中的信息压缩到载体图像中最不明显的部分。提出系统的架构如图1所示。

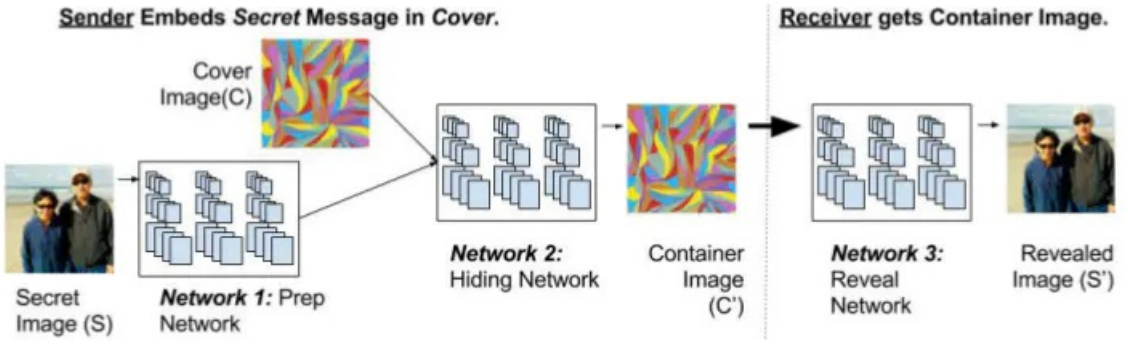

Figure 1: The three components of the full system. Left: Secret-Image preparation. Center: Hiding the image in the cover image. Right: Uncovering the hidden image with the reveal network; this is trained simultaneously, but is used by the receiver.

Fig1.png

图1所示的三个组件被训练为一个单独的网络；但是，最容易单独描述它们。最左边的预备网络准备隐藏加密图像。此组件有两个用途。首先，在加密图像（尺寸MM）小于载体图像（NN）的情况下，预备网络将加密图像的大小逐步增加到覆盖图像的大小，从而将保密图像的比特位分布到整个NN像素上。（由于篇幅原因，作者不提供小图像实验的细节，而是集中在全尺寸图像上）。与所有尺寸的隐藏图像相关的更重要的目的是将基于颜色的像素转换为更有用的特征，以便简洁地编码图像，例如边缘[20，21]，如图2所示。

从左数第二个/主网络，即隐藏网络，将预备网络和载体图像的输出作为输入，并创建容器图像。该网络的输入是一个NN像素场，载体图像的深度是RGB三通道与加密图像的特征通道数相加。我们的研究尝试了30多个隐藏层和卷积大小不同的网络架构；其中最好的包括5个卷积层，每个卷积层有50个过滤器，每个过滤器有33、44、55块。最后，最右边的网络，显示网络，被图像的接收器使用；它是解码器。它只接收载密容器图像（而不是载体或加密图像）。译码器网络去除载体图像，显示加密图像。

如前所述，我们的方法很大程度上借鉴了自动编码网络[22]；但是，我们不是简单地通过瓶颈对单个图像进行编码，而是对两个图像进行编码，使中间表示（载密容器图像）尽可能与载体图像相似。系统通过减少以下误差进行训练（其中的c和s分别表示载体cover图像和加密secret图像，和分别表示stego图像和恢复出的secret图像，是用来权衡重构误差）：

重要的是要注意计算错误的位置以及每个错误影响的权重，可以看到显示网络Reveal Network不受的影响，而部分的误差会反向传播给三个部分网络的所有参数，请参见图3。特别注意，错误项不适用于接收载密容器图像并提取秘密图像的显示网络的权重。另一方面，所有的网络都接收到错误信号用于重建隐藏图像。这样可以确保在准备网络早期形成的表示以及用于覆盖图像重建的表示也对秘密图像的信息进行编码。

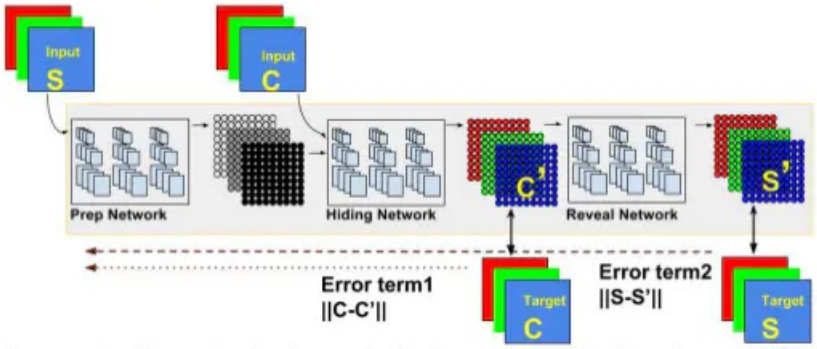

Figure 3: The three networks are trained as a single, large, network. Error term 1 affects only the first two networks. Error term 2 affects all 3. S is the secret image, C is the cover image.

Fig3.png

为了确保网络不是简单地对最低有效位LSB中的秘密图像进行编码，在训练期间，在第二个网络的输出（例如，在生成的载密容器图像中）中添加少量噪声。噪声的设计使得LSB偶尔会翻转；这确保LSB不是秘密图像重建的唯一容器。稍后，我们将讨论秘密图像信息的放置位置。接下来，我们将研究网络在实践中的表现。

3.实证评价

使用Adam[23]对这三个网络进行了如上所述的训练。为了简单起见，重构图像最小化了像素差的平方和误差，尽管其他图像度量可以很容易地替换[24，25]。使用从ImageNet训练集中随机选择的图像对网络进行训练[26]。

定量结果如图4所示，由每个像素、每个通道的SSE（误差平方和）测量。测试是在从ImageNet图像（没有训练使用的）中获取的1000对图像上进行。为了进行比较，还显示了使用相同的网络只对没有秘密图像的载体图像进行编码的结果（例如）。这提供了使用该网络对载体进行最佳重建的错误（这在编码秘密图像的同时是无法实现的）。图4中还显示了覆盖和重建的误差柱状图。可以看出，大像素的错误是很少的。

	β	Cover	Secret
Deep-Stego	0.75	2.8	3.6
Deep-Stego	1.00	3.0	3.2
Deep-Stego	1.25	6.4	2.8
Cover Only	0.00	0.1	(n/a)

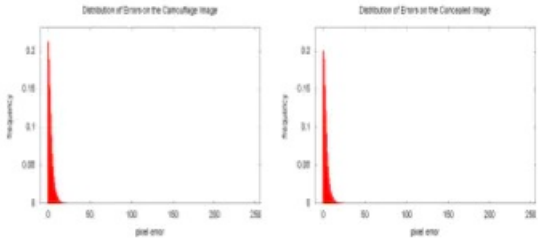

Figure 4: Left: Number of intensity values off (out of 256) for each pixel, per channel, on cover and secret image. Right: Distribution of pixel errors for cover and secret images, respectively.

Fig4.png

图5显示了隐藏六个图像的结果，选择这些图像是为显示不同的错误率。这些图像不是从ImageNet中获取的，以证明网络没有过度训练成ImageNet数据集的特性，并且可以工作在使用手机摄像头和DSLR拍摄的一系列照片上。请注意，大多数重建的载体图像看起来几乎与原始载体图像相同，尽管编码所有信息以重建秘密图像。原始图像和载体图像之间的差异显示在最右边的列中（强度放大5倍）。

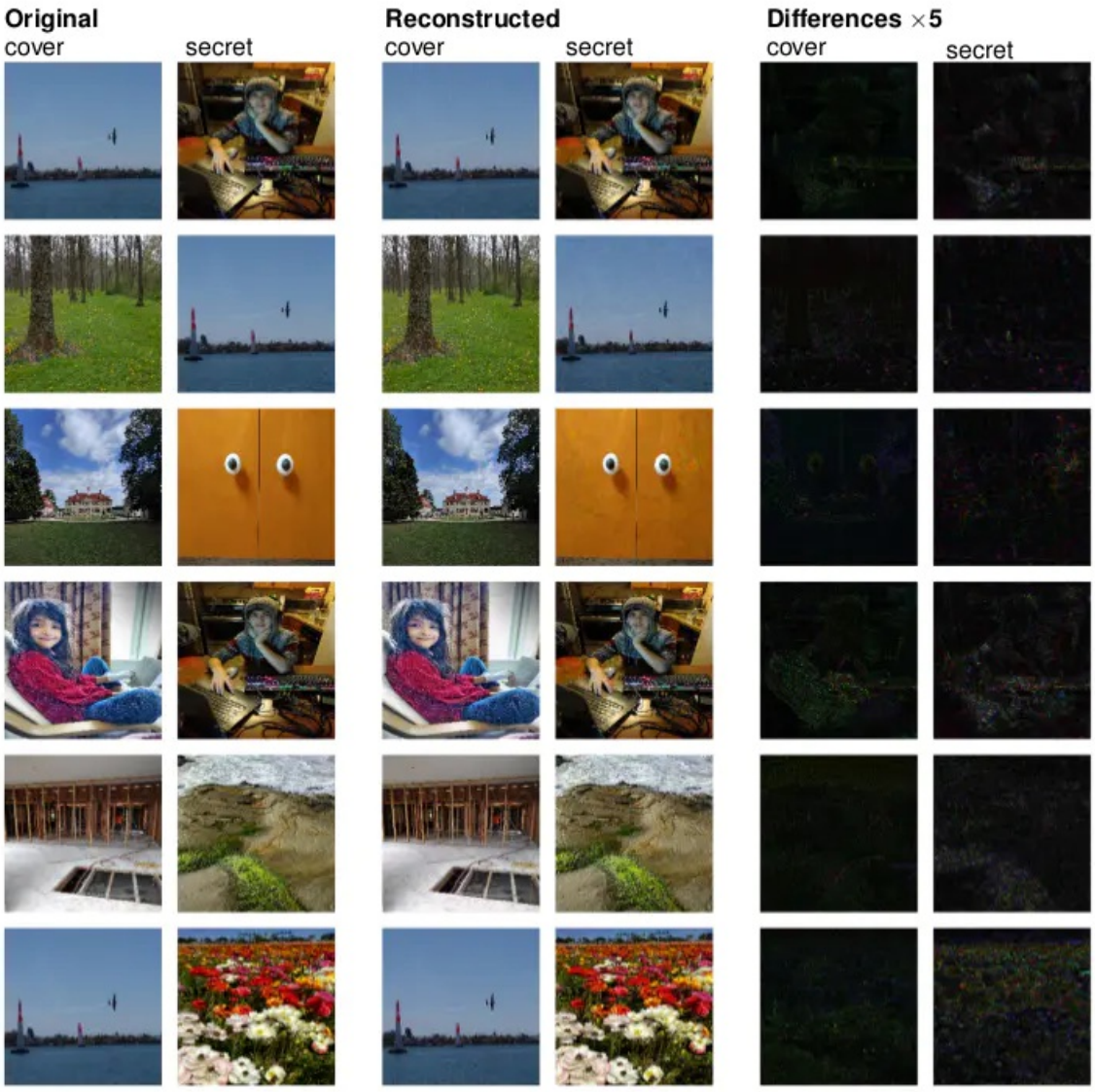

Figure 5: 6 Hiding Results. Left pair of each set: original cover and secret image. Center pair: cover image embedded with the secret image, and the secret image after extraction from the container. Right pair: Residual errors for cover and hidden – enhanced 5×. The errors per pixel, per channel are the smallest in the top row: (3.1, 4.5) , and largest in the last (4.5, 7.9).

Fig5.png

下面考虑如何将模型的错误率和简单的LSB隐写的错误率比较，论文设计了如下的内容：使用加密secret图像的最高4比特（MSB）来替换载体cover图像的最低4比特（LSB）。在这个过程中，为了恢复出加密secret图像，我们从载密容器container图像中复制加密secret图像的最高4比特，剩下的4比特设置为整个数据集的平均值。在这种情况下，载体cover图像在每个通道上的平均像素误差为5.4（范围为0-255），加密secret图像的平均误差（使用丢失LSB位的平均值时）约为4.0。载体cover图像的重构误差高于4.0，这就说明了自然图像的比特位分布是不同于MSB和LSB的，因此即使载体cover图像和加密secret图像符合相同的分布，当使用secret图像的最高4比特位替换cover图像的最低4比特位时，会产生比使用最低4比特位的平均值替换时更高的误差。更重要的是，这时的误差时明显高于Figure 4系统实现的误差。

（请注意，当使用平均值填充LSB时，预期误差为4.0：从像素的编码中删除4位，可以表示的强度将减少16倍。通过选择平均值来替换丢失的位，最大误差为8，平均误差为4，假设位均匀分布。为了避免混淆，我们指出，虽然也考虑使用封面图像的平均值，但请记住载体图像的LSB是存储加密图像的MSB的位置。因此，这些位必须在这个编码方案中使用，因此误差更大。）

我们在结束本节时演示了我们方法的局限性。回想一下，这些网络是根据ImageNet数据集中的自然图像进行训练的。虽然这覆盖了非常大的图像范围，但是当使用其他类型的图像时，检查效果是很有启发性的。尽管该数据集中包含了大量的图像，但我们应该尝试使用其它类型的图像来进一步地检验模型的效果。图6显示了五幅这样的图像。在第一行中，使用纯白色图像作为载体，检查隐藏彩色加密图像的视觉效果。在使用ImageNet图像进行训练时未遇到此简单情况。第二行和第三行将加密图像更改为亮粉色圆圈和均匀的噪声。可以看出，即使载密容器图像（第4列）只包含有限的噪声，恢复的加密图像也是非常嘈杂的。最后两行中，载体图像分别变为圆形和均匀噪声。正如预期的那样，虽然加密图像仍然可以识别，但是现在重建载体和加密的误差是很大的。尽管此时secret图像仍然能够提取出来，且仍是可识别的，但此时的误差已经变得很大。

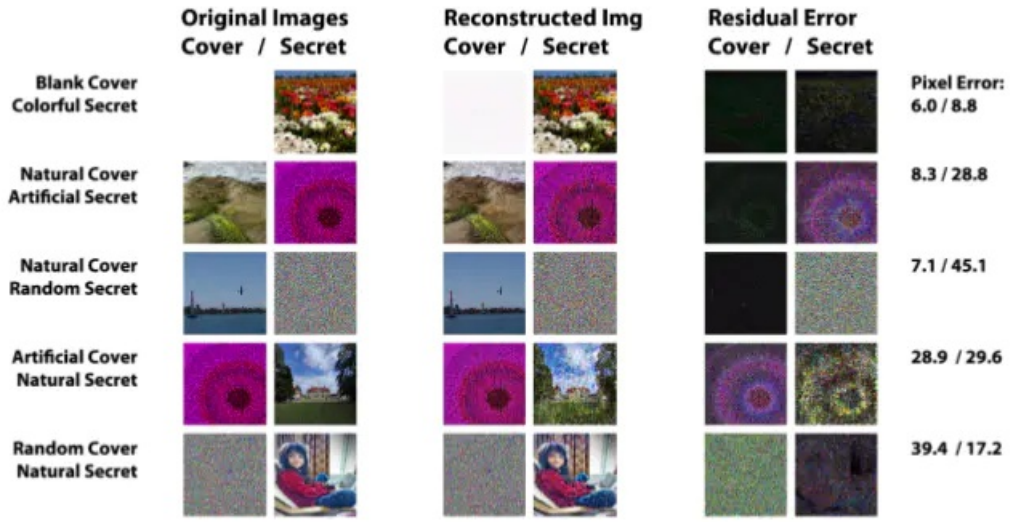

Figure 6: Results with images outside the set of natural images.

Fig6.png

3.1 如果原始的载体图片变得可以访问该怎么办？

对于许多隐写术应用场景，可以安全地假设攻击者无法访问原始载体图像（不嵌入加密图像）。但是，如果原始载体图片被发现了呢？那么，即使没有进入解码网络也能确定图像中是否还有加密图像的信息？在图5中，我们展示了原始载体和具有5x增强功能的载密容器之间的差异图像——几乎看不到任何东西。我们重新检查了5x（5倍）、10x和20x增强时的残差图像（在适当的情况下剪切为255）见图7。在第一行中，请注意残余图像（20倍）与载体图像非常相似，可以明显的看到原始载体cover图像的轮廓。在第二行中，残差部分是载体和加密图像的组合，20倍残差图像中可以看到的是原始载体cover图像和加密secret图像轮廓的一个重叠。在第三行中，我们看到了最麻烦的结果——加密图像的特征被显示出来，20倍残差图像已经可以明显辨别出载密stego图像的特征。（请记住，只有当原始未扰动图像可供比较时才会发生这种情况）。有许多标准的模糊处理方法，例如在将加密图像嵌入到载体图像之前添加加密编码。为了避免这种情况，我们可以采取一些典型的方法，如在隐写之前先将secret图像进行加密。作者演示了另一种可以与任何其他方法结合使用的方法：修改网络的误差函数。

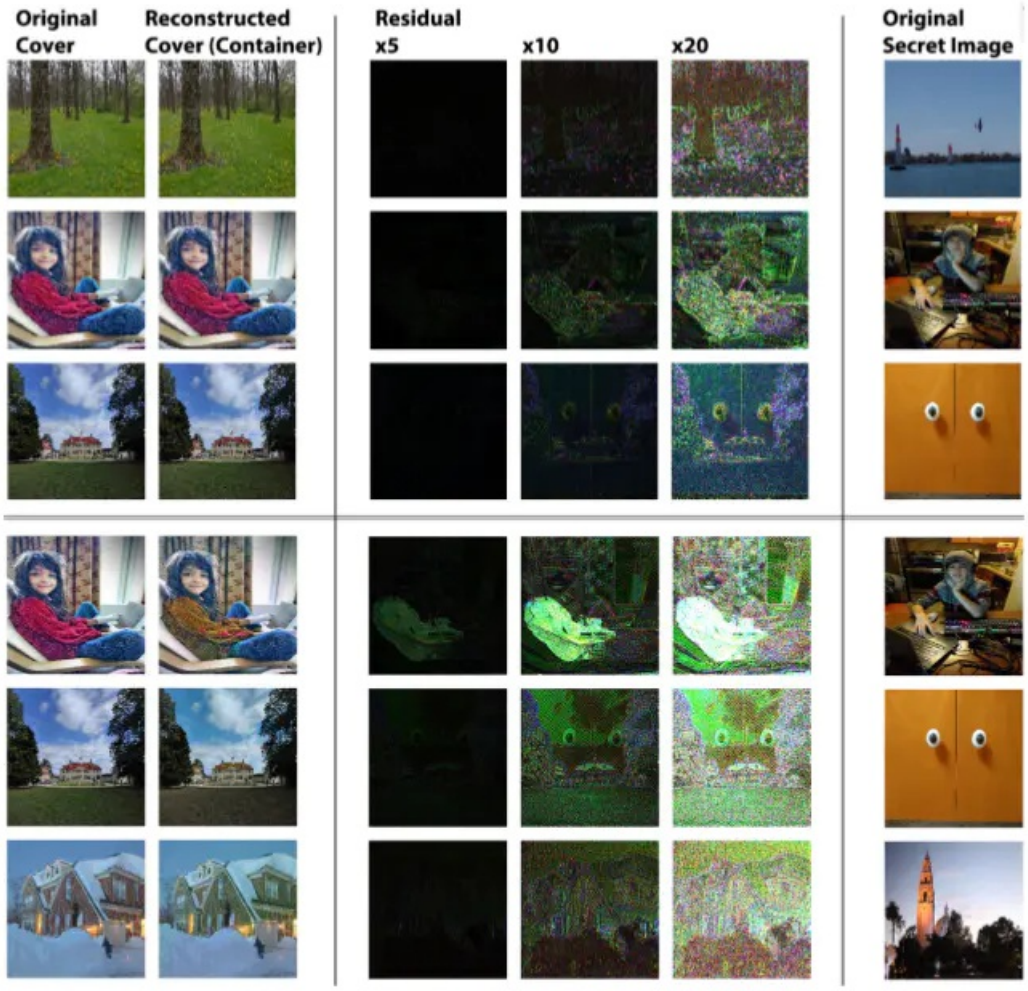

Figure 7: Top 3 rows. If the original image is leaked and is subtracted from the container image, the residual can be computed. With enough enhancement (20x), some of the secret image is revealed. Bottom 3 rows: by explicitly creating an error term that minimized the correlation between the residual and the secret image, the residual reveals less about the secret image; however, the pixel errors for the container rise (note the less saturated colors in some of the red regions).

Fig7.png

除了所描述的两个误差项外，作者还添加了一个误差项，该误差项最小化了载体图像残差与加密图像之间的像素相关性，其中，S是加密图像。这个项的许多权重都经过了实证检验。在图7（底部）所示的结果中，它被缩放到大约（0.15像素数通道数）。将残差与秘密图像的相关性最小化会从残差中去除秘密图像的许多特征——即使使用了20倍的放大倍率。当然，鲁棒性和恢复力是有代价的；如图所示，一些重建的质量已经降低了（见第一幅图像中红色的饱和度）。 这一误差项的权重是由经验决定的，Figure 7的下半部分加入了这一误差项，效果比上边的要好了一些。

4.加密图像在哪里编码？

本文的主要目的是具体地证明，在图像中对大量信息进行编码是可能的，并且视觉上可察觉的部分是有限的。然而，还没有明确的尝试来主动地隐藏机器检测中存在的信息。虽然我们不能期望完全隐藏这样一个事实，即高达1/2的信息是隐藏信息的一部分，但可以采取措施使其更难被发现。然而，首先，我们必须确定秘密图像的信息存在哪里。

网络仅仅是把加密图像的信息隐藏在载体图像的最低有效位吗？存在用于查找LSB中隐藏信息的工具。这样一个公开可用的隐写分析工具包，StegExpose，被用来测试我们隐藏图像的可检测性[27-29]。根据该工具的描述：“Stegexpose评级算法是由预先存在的基于像素的隐写分析方法（包括Dumitrescu（2003年）的样本对、Fridrich（2001年）的RS分析、Westfeld（2000年）的卡方攻击和Dumitrescu（2002年）的主集）的智能和彻底测试组合而来的”[27]。除了默认设置（阈值=0.2），检测阈值在很大范围内也会发生变化。StegExpose的ROC曲线如图8所示。注意小的变化超出随机猜测（绿线）。

论文使用StegExpose检测了模型产生的stego图像，得到的ROC如Figure 8所示，这就证明了模型不是简单的将秘密信息隐写到LSB。论文中还尝试去对stego图像RGB三个通道中的像素值进行扰动，观察对提取出来的secret图像的影响，结果如Figure 9所示。可以看到，stego中RGB任何一个通道的扰动都会影响到secret图像中RGB三个通道中的像素点，这就证明了模型是将秘密信息隐写到了三个通道中。

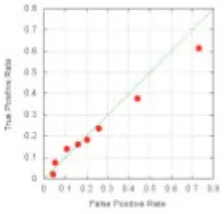

Figure 8: ROC curves: True Positive Rate vs. False Positive Rate for **StegExpose** when trying to detect images embedded via the proposed method.

Fig8.png

如果只把它放在LSB位中，StegExpose应该能够找到信息。我们使用第二种方法来查找存储信息的位置。研究中使用的图像在每个像素上由24位组成。如果我们翻转载密容器图像中所有像素的R通道的第一位，我们可以测量它对载体容器图像自身重建的影响，也可以通过显示网络传播修改后的图像，对秘密图像的重建。效果非常显著，见图9。

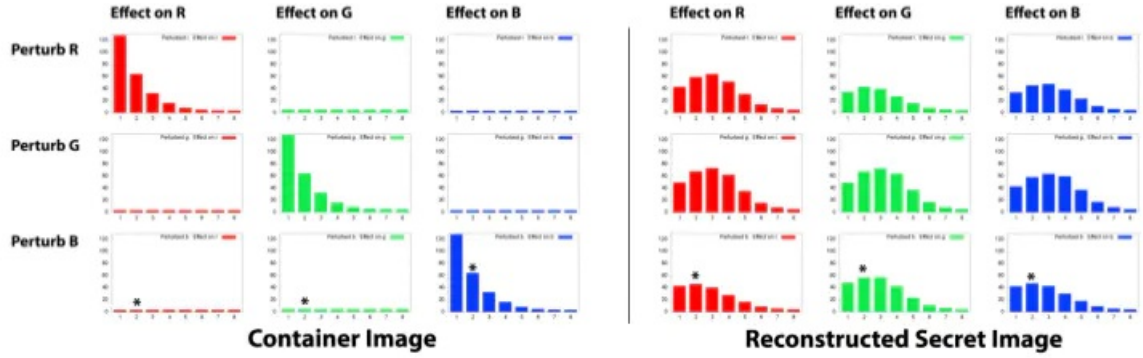

Figure 9: Sensitivity to Bits in the Container Image. (Left:) Control case, the container image. Changing a bit in the container image obviously has no effect on other bits. (Right): However, that same bit flip in the container image has effects across all color bands in the recovered secret image. For example, if we want to measure the effect of flipping the 2nd bit in the Blue channel, we would look at the marked entries (marked with *).

Fig9.png

在图9中，图的左半部分应被视为“控件”。左上角的条形图显示，在容器的红色通道中更改位的效果只对容器中的红色通道有影响，并且大小与位的重要性成正比；这与预期完全一样，对于绿色通道和蓝色通道也是如此（如对角线所示）。更有趣的是图9的右侧。在这里，我们看到在容器图像的任何颜色通道中，任何位位置的翻转都会对秘密图像重建中的所有颜色通道产生影响。此外，该误差不遵循位位置的标准显著性排序。换句话说，秘密图像的信息在彩色通道中传播，因此之前没有检测到它的原因。载密stego中RGB任何一个通道的扰动都会影响到加密secret图像中RGB三个通道中的像素点，这就证明了模型是将秘密信息隐写到了三个通道中。

尽管StegExpose没有发现编码，但我们怀疑这一数量的信息应该很容易被发现。最近在多项研究中已经证明，深度网络具有竞争力，并且通常比使用手工选择图像特征的更成熟的隐写分析方法要好[12–14]。有趣的是，许多已发表的隐写分析技术并不是盲目的：它们假定仅从已知的隐藏方法检测隐藏图像，甚至可以访问有关载体图像分布的统计数据。这使得隐写分析任务更容易，而隐藏任务更困难。我们不会在这里讨论这个假设的现实局限性，但会遵循这个例子。

我们训练隐写分析网络作为二元分类器，使用未扰动的图像网络图像作为负样本，并以载密容器作为正样本。我们发现二进制输入层的隐写分析网络性能良好。24层中的每一层在输入图像的像素表示中表示一个位。在这项研究中，网络以类似于其他网络的方式进行了训练；再次尝试了多种架构和学习速度。

这个完全卷积网络能够获得90-95%的综合分类率。为了进行比较，对相同的网络进行了重新训练，以找到最低有效位替换（其中载体图像中每个颜色通道的L最低有效位替换为加密图像中的L最高有效位）。对这些网络进行了训练和测试，L=1、2、3、4；所有这些都达到了99%以上的综合比率。另一组测试表明，即使将单个位随机放置在4位位置中的一个，检测率仍超过99%。考虑到这些检测率即使是一个比特，高检测率与一个完整的图像是不足为奇的。设计了隐写分析网络来验证模型的抗检测性，该模型的输入是，24个层中每一层是图像像素的24个比特位中的一位，最终的检出率在90-95%。作为对照，同样的模型在采用LSB隐写的图像上做训练，最终的检出率超过了99%。当然这只是个补充实验，不能否定模型的有效性及意义。

尽管如此，这为今后的工作指明了直接的方向。本研究在建立完整的编码系统后，对隐写分析网络进行训练和使用。然而，在一个对抗性学习框架中使用预先训练和/或同时训练的隐写分析网络，例如生成对抗性网络(GAN)[30]，在隐写网络的训练过程中提供了一种方法来合并基于混淆的错误度量。敌手根据隐藏消息的可发现性提供一个补充错误信号，该信号在重构错误之外被最小化。[31]最近成功地探索了一个非常类似的过程，其中包含隐藏在图像中的小消息。

5.讨论和后续工作

在本节中，我们简要讨论了本研究中发现的一些观察结果，并提出了后续工作的想法。首先，让我们考虑培训网络的可能性，以便在部署系统之后恢复隐藏图像，并且不访问原始网络。可以想象，如果攻击者能够获得目标系统创建的载密容器图像的多个实例，并且在每个实例中，如果还提供了两个组件图像（载体或加密图像）中的至少一个，则可以训练网络来恢复两个组件。在没有访问真值“训练”数据的情况下，攻击者能做什么？使用平滑度约束或其他来自更经典图像分解和盲源分离的常见启发式方法[32–34]可能是第一种选择。通过这些方法中的许多，即使获得少量的训练数据也将有助于调整和设置参数和优先级。如果预计会发生此类攻击，则可以进一步研究在多大程度上适应第3.1节所述的技术可能会降低这些尝试的有效性。

如前一节所述，在当前形式下，正确检测隐藏图像的存在（不一定是确切的内容）确实是可能的。与载体图像的数据相比，隐藏的信息量（1:1）使发现率很高。这远不止是最先进的传输可靠未检测到的信息的系统。提出了一种通过显式降低载体图像残差与隐藏图像的相似性来提高隐藏图像内容恢复难度的方法。虽然超出了本文的范围，但是我们可以通过以下补充机制使系统具有更大的弹性。在隐藏秘密图像之前，像素会按照之前商定的M种方法中的一种进行排列(到位)。然后，置换的加密图像被系统隐藏，密钥也被隐藏(索引到M中)。这使得即使通过查看残差(假设可以访问原始图像)也很难恢复，因为残差没有空间结构。这种方法的使用必须与（1）发送置换密钥的需要（尽管可以在几个字节内可靠地发送）和（2）置换后的秘密图像实际上更难编码，从而可能增加整个系统的重建错误。最后，需要注意的是，为了使用这种方法，本研究中经过训练的网络必须经过再训练才能使用。由于隐藏网络无法利用秘密图像中的局部结构对信息进行编码，必须对整个系统进行再训练。

这项研究为隐写术的探索开辟了一条新的途径，更广泛地说，在图像中放置补充信息。以前有几种方法试图使用神经网络来增强或替换图像隐藏系统的一小部分。我们已经演示了一种创建完全可训练系统的方法，该系统可以在不引人注目地将一个完整大小的彩色图像放入另一个图像中时提供非常好的视觉效果。虽然系统是在图像的上下文中描述的，但是可以训练同一系统嵌入文本、不同大小的图像或音频。此外，通过使用音频文件的频谱图作为图像，这里描述的技术可以很容易地用于音频样本。

扩大这项工作有许多直接和长期的途径。其中最直接的三个列在这里。（1）为了建立一个完整的隐写系统，应该解决对统计分析者隐藏信息存在的问题。这可能需要一个新的训练目标（例如对手），也可能需要在大的覆盖图像中编码较小的图像。（2）本文所述的嵌入方案不用于有损图像文件。如果需要有损编码，例如jpeg，那么可以直接使用DCT系数而不是空间域[35]。（3）为了简单起见，我们使用了一个简单的SSE误差度量来训练网络;然而，与人类视觉更密切相关的错误度量，如SSIM[24]，可以很容易地替换。