于众目睽睽之下隐藏图像：深度隐写术

翻译

Marcovaldo 于 2017-12-07 16:18:31 发布 10235 收藏 29

分类专栏： 深度学习 隐写 文章标签： 深度学习 隐写


深度学习 同时被 2 个专栏收录

6 篇文章 0 订阅

订阅专栏


隐写

1 篇文章 0 订阅

订阅专栏

博客首发至Marcovaldo’s blog (http://marcovaldong.github.io/)

今天要介绍的是Google Research在NIPS 2017上发表的一篇论文，它的主要工作是将深度学习应用于图像隐写中，实现了在图像中隐写另一张图像。下面具体介绍一下这篇论文做了哪些工作。

文章首先介绍了什么是隐写术及隐写术的应用，这里不再赘述。我们将载体成为cover，载密图像称为stego。因为在图像中嵌入秘密消息会改变图像载体的视觉外观和基本的统计信息，因此隐写的两个关键点是隐写量和载体本身。常见的隐写是将文本信息隐藏到图像中，因此衡量隐写量的基本单位是比特每像素（bits-per-pixel, bpp），通常情况下的隐写量设置为0.4bpp或者更低。嵌入秘密信息越长，隐写量越大，stego被识别出来的可能性越大。另一个影响隐写的是载体本身。我们通常将秘密信息隐写在纹理复杂、充满噪声、高频率的图像区域，这样对图像的视觉外观改变较小，不易引起人们的注意。

近几年已经有工作将深度学习应用于隐写分析，如前面两篇博客中介绍的几篇论文，但少有工作关注隐写本身。这篇论文使用神经网络来判断在图像的那些位置嵌入秘密信息，训练出一个编码器来将秘密信息嵌入，使得秘密信息能够分散在图像的每一个比特位中，而不是隐写在某一个LSB上。同时，模型还训练一个解码器，能够从stego中将secret图像提取出来。文章的目标是将一个NxNxRGB大小的secret图像隐写到同样大小的载体图像cover中，且对cover的扰动尽可能的小。这样以来，就放松了之前的secret信息必须要无损的重构出来的限制，可以在可接受的stego的质量和secret图像的质量间有一个折中。现有的隐写分析方法可以检测到隐写率低至0.1bpp的stego，本文中的隐写率比之前的方法要高十几倍甚至40倍。尽管视觉上很难被察觉，但由于隐藏的信息量很大，这种方法下产生的stego被检测出的概率肯定会不小。论文会在后面证明这种方法不会被常用的简单隐写分析方法击破，并给出了如何根据要求在抗检测和secret图像的恢复质量之间进行折中的可以尝试的方向。

模型要学习的是如何对secret图像进行压缩，并将压缩后的信息嵌入到cover的最不易察觉的部分。整个模型的结构如Figure 1所示。最左侧是Pre-Network，对secret图像进行预处理，这部分主要由两个作用：一是secret图像的尺寸可能会小于cover，因此模型会将原来MxM的secret图像的比特分布到NxN（cover的尺寸）个像素点上来。受篇幅限制，论文没有提供如何将小尺寸的secret图像进行放大，集中讨论的是secret图像与cover相同尺寸的情况。更重要的一点是将原图像的基于色彩的像素点转换成方便对图像进行编码的更有价值的特征，如Figure 2中的边缘信息等。

模型的第二部分Hiding Network将Prep-Network的输出和cover图像作为输入，输出的是stego图像（文章中使用的词汇是Container）。输入的尺寸是NxN，深度是RGB三通道加上前面提取到的secret图像的特征通道数。在30多个模型结构中，最好的结构是5个卷积层，每个卷积层包含50个卷积核，卷积核的尺寸选择的是3x3、4x4、5x5。Figure 1中最右侧的是提取模型Reveal Network，它作为一个解码器，输入的是Hiding Network输出的stego图像，输出的是恢复出的secret图像。

如前面所述，本文中的方法主要借助于自动编码网络，对两个图像进行编码使得产生的stego图像与cover图像尽可能的相似。这里使用的是下面的损失函数，其中的c和s分别表示cover图像和secret图像， c

s

可以看到Reveal Network不受 c

训练过程中使用的优化器是Adam，简单起见最小化的是像素间的平方和误差（当然，这里也可以使用其他的metric），使用的图像是从ImageNet数据集中随机选择的图像对。Figure 4中展示的是每通道每像素上的SSE。这个数据是在未参与训练的1000对图像上的测试结果。另外还给出了cover图像和stego图像的像素差值的一个直方图，可以看到几乎没有大的像素差值。Figure 5展示的是隐藏6张图像的效果及误差率的一个直观展示。这里使用的图像不是ImageNet数据集中的，而是使用手机或单反拍摄的。可以看到，尽管嵌入一张secret图像，重构出的cover图像也就是stego几乎和原cover图像是相同的。

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secret图像的最高4比特来替换cover图像的最低4比特。在这个过程中，为了恢复出secret图像，我们从container图像中复制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，cover图像在每个通道上的平均像素误差为5.4，secret图像的平均误差接近4.0。cover图像的重构误差高于4.0，这就说明了自然图像的分布式不同于MSB和LSB的，因此即使cover图像和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中的误差的。

前面提到，模型在训练时是随机从ImageNet数据集中选取的图像对，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。Figure 6中给出了模型在5组图像上的效果，可以看到尽管此时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。

在很多隐写场景中，我们通常假定攻击者是无法获取到原始的cover图像的，但是如果原始图像是可以获取到的呢，在没有解码器的情况下能否确定图像中是否还有secret图像呢？Figure 5中对cover图像和stego图像中的误差进行了5倍的增强，能看到有一些噪点，但是看不出任何有信息的东西。Figure 7中对误差分别进行了5倍、10倍和20倍的增强。可以看到，第一行中的20倍残差图像中可以明显的看到原始cover图像的轮廓，第二行中的20倍残差图像中可以看到的是原始cover图像和secret图像轮廓的一个重叠，第三行中的20倍残差图像已经可以明显辨别出stego图像的特征。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。另外，我们可以在误差函数中加入一个误差项 cover图像和stego图像之间的残差与

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。

另外，论文中还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24x(NxN)，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。

最后，论文给出了三个方向：一是构建更完整的隐写模型，增强抗检测能力，并指出有必要设计新的目标函数，或在大尺寸的cover图像中隐写入小尺寸的秘密图像；二是本文中的方法不适用于有损图像，在有损图像中或许可以直接在DCT域而不是在空域中做文章；三是在训练模型时，论文中直接使用了SSE作为metric，或许应该使用跟人类视觉更相关的metric，如SSIM。

附录

[1] Shumeet Baluja, Hiding Images in Plain Sight: Deep Steganography, NIPS 2017