

图像信息隐写相关论文四

原创

[ningchewinbell](#)



于 2020-04-20 11:54:10 发布



991



收藏 12

分类专栏: [图像](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/ningchewinbell/article/details/105631197>

版权



[图像](#) 专栏收录该内容

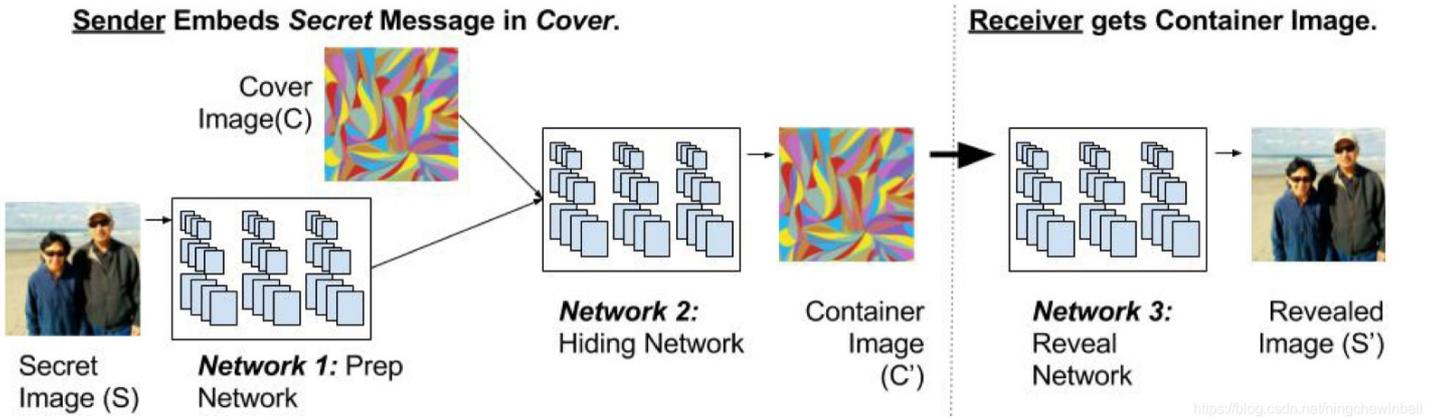
8 篇文章 2 订阅

订阅专栏

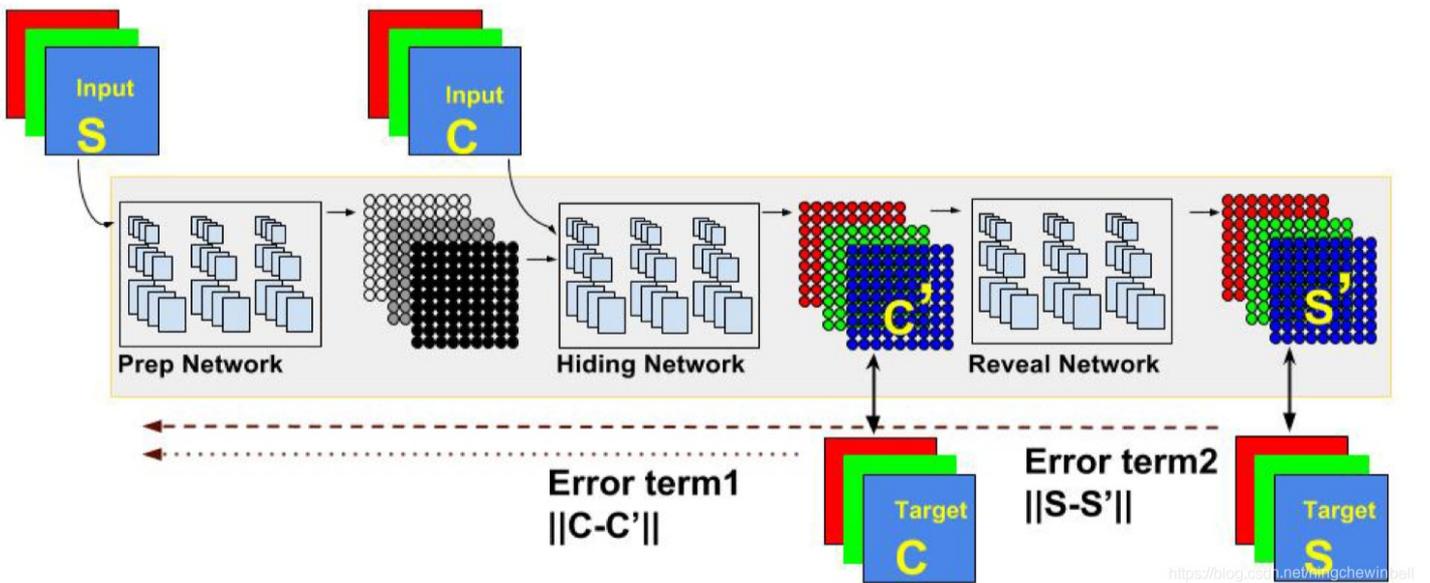
今天主要整理三篇图像隐写的文章, 即在cover image中隐写大小与cover image一样的payload image。

[Hiding Images in Plain Sight: Deep Steganography \(2017\)](#)

这篇文章目标是在另一个 $N \times N \times \text{RGB}$ 的cover image中隐藏完整的 $N \times N \times \text{RGB}$ 像素秘密图像，而对cover image的失真最小（每个颜色通道为8位）。但是，与以前的研究（必须以完整的重构发送的隐藏文本消息）不同，这里放宽了无损接收秘密图像的要求。并且在载体和秘密图像的损失上找到可以接受的折中。



整个网络架构由三部分组成：**Prep-Network**，用于将secret image编码；**Hiding Network**，将Prep-Network的输出和cover image作为输入，输出Container image，完成隐写；**Reveal Network**，他是接受者使用的网络，可以从Container image中提取出secret image。

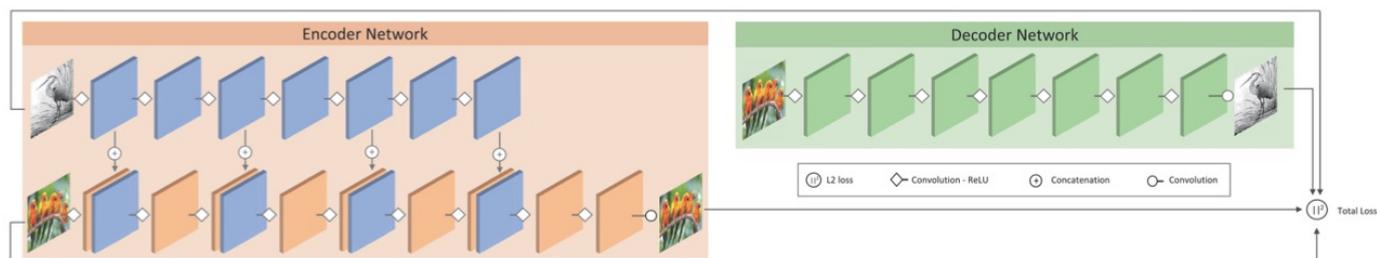


损失函数也比较简单，就是cover image隐写了信息后得到的Container image与cover image的像素平方差，以及secret image与接收方解码得到的secret image的像素平方差。

End-to-End Trained CNN Encoder-Decoder Networks for Image Steganography (2018)

这篇文章提出了一个基于CNN的编码解码模型来实现图像隐写（将一张灰度图像隐写到大小相同的彩色图像中）。

主要贡献：1.提出了一个基于CNN的编码解码结构的隐写模型；2.介绍了一个可以实现端到端训练编码解码结构网络的损失函数；3.在一系列具有挑战性的数据集上进行了实验，报告了PSNR和SSIM值。



<https://blog.csdn.net/ningchewinbell>

具体来说，编码器包括两个分支：Guest分支和Host分支，Guest分支的输入为灰度秘密图像（ I_g ），由卷积和ReLU层组成；Host分支的输入为cover图像（ I_h ），同样由卷积和ReLU层组成（最后一个卷积没有ReLU激活）。编码器中Guest分支的特征会拼接到Host分支，如上图所示，最后输出隐写图像 O_e 。

解码器同样由卷积和ReLU层组成（最后一个卷积没有ReLU激活）。输入为编码器的输出 O_e ，输出恢复的秘密图像 O_d 。

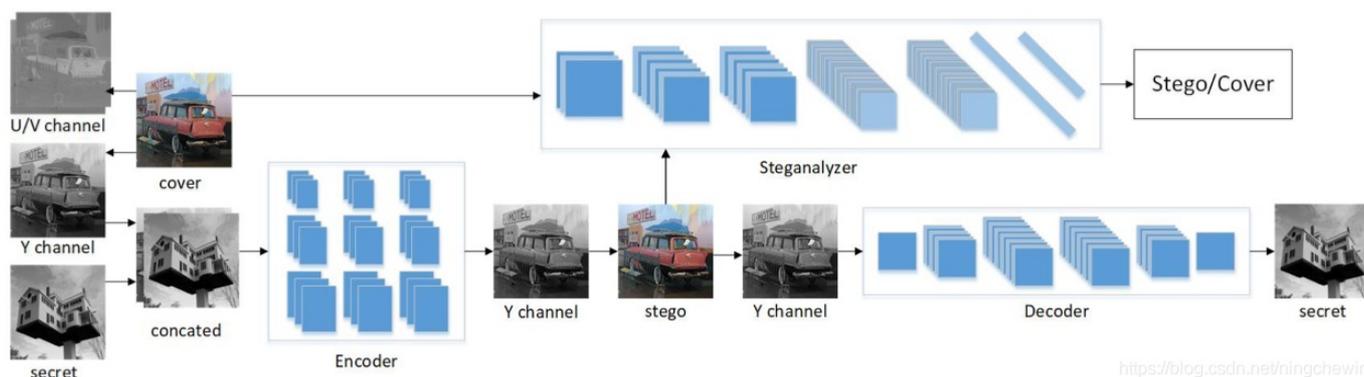
训练的损失函数为：

$$L(I_g, I_h) = \alpha \|I_h - O_e\|^2 + \beta \|I_g - O_d\|^2 + \lambda (\|W_e\|^2 + \|W_d\|^2)$$

Invisible Steganography via Generative Adversarial Network (2018)

针对现有许多基于深度学习的隐写算法隐写后图像颜色发生改变的情况，这篇文章提出了一个基于CNN的隐写模型（ISGAN）将一张灰度图像隐写到一张彩色图像中，然后在接收端能将秘密图像恢复出来。

文章的主要贡献：1.通过仅在cover image的Y通道隐写秘密图像来改善不可见性；2.引入了生成对抗网络，通过最小化隐写图像和自然图像的经验概率分布之间的差异来增强；3.构建了一个混合损失函数，以生产更符合人类视觉的图像。实验表明在LFW、PASCAL-VOL12和ImageNet数据集上都更优。

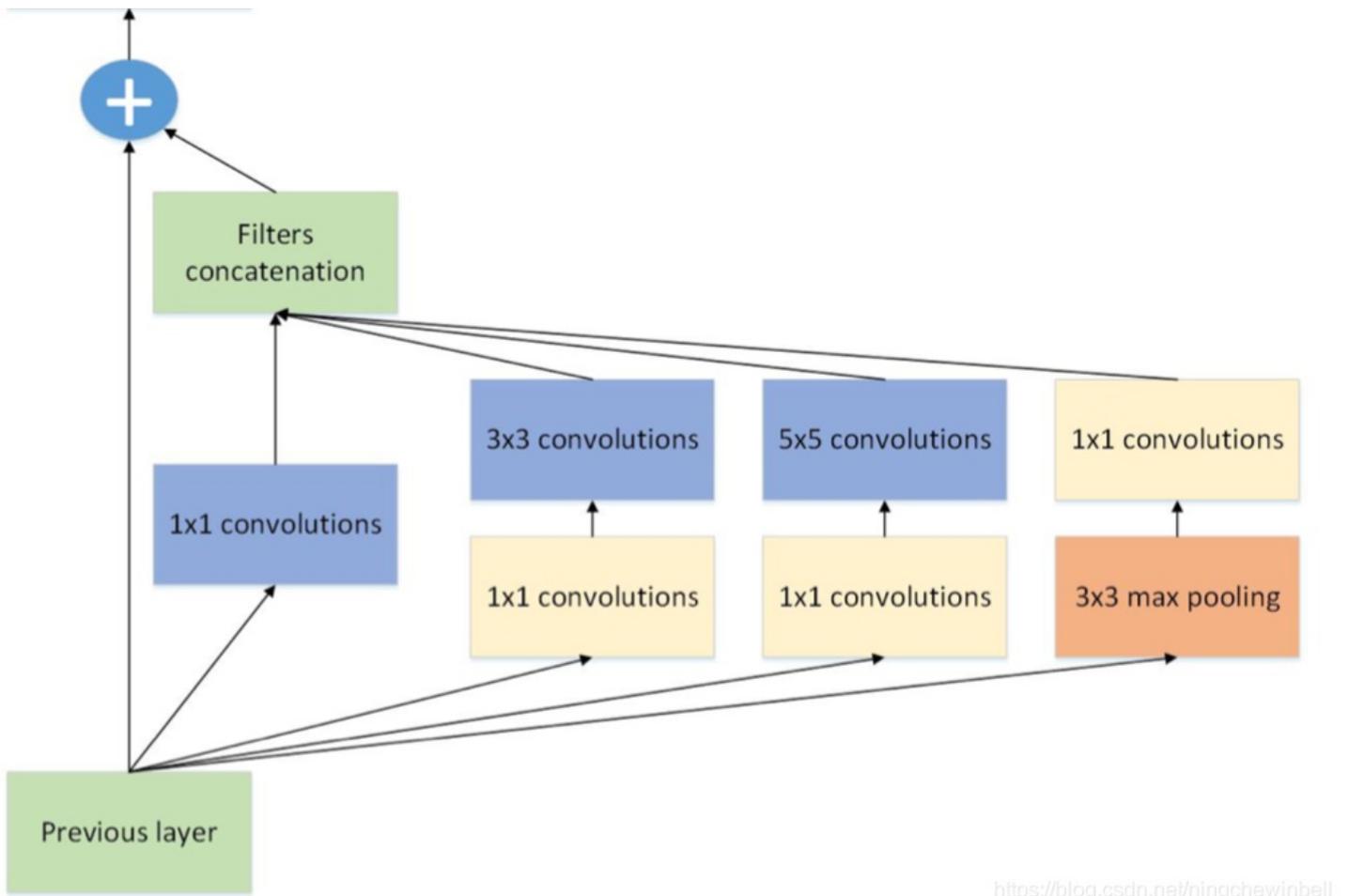


<https://blog.csdn.net/ningchewinbell>

RGB色彩空间的彩色图像每个通道都包含语义信息和色彩信息，当转换成YCrCb色彩空间时，Y通道只包含语义信息，Cr和Cb包含部分语义信息和所有色彩信息。

Basic Model为上图的编码器加解码器部分（就是不包含分析器），其中编码器用到了图像分类效果很好的inception模块，详细结构文中用表格列出来了所有层。解码器是全卷积网络，具体信息也在文中列了表格。

ReLU activation



<https://blog.csdn.net/ningchewinbell>

上图是本文用到的inception模块，加入了残差网络。

隐写分析器的具体结构也在文中以表格的形式列出来了，本文的损失除了MSE外，还用到了SSIM、MSSIM。

$$L(c, c', s, s') = \|c - c'\| + \beta \|s - s'\|$$

$$L(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

$$C(x, y) = \frac{2\theta_x\theta_y + C_2}{\theta_x^2 + \theta_y^2 + C_2}$$

$$S(x, y) = \frac{\theta_{xy} + C_3}{\theta_x\theta_y + C_3}$$

$$SSIM(x, y) = [L(x, y)]^l \cdot [C(x, y)]^m \cdot [S(x, y)]^n$$

<https://blog.csdn.net/ningchewinbell>

$$\begin{aligned} L(c, c') &= \alpha(1 - SSIM(c, c')) \\ &+ (1 - \alpha)(1 - MSSIM(c, c')) \\ &\quad + \beta MSE(c, c') \end{aligned}$$

$$\begin{aligned} L(s, s') &= \alpha(1 - SSIM(s, s')) \\ &+ (1 - \alpha)(1 - MSSIM(s, s')) \\ &\quad + \beta MSE(s, s') \end{aligned}$$

$$L(c, c', s, s') = L(c, c') + \gamma L(s, s')$$

<https://blog.csdn.net/ningchewinbell>

最后，实验部分包括：自身损失函数的对比实验（PSNR作为评估指标），与上一篇文章的对比实验（分别在两个数据集上，LFW和Pascal VOC2012，对比实际图片效果）