图像隐写分析——生成含密图像 Stego Image

原创

DRZ_2000 于 2021-07-09 13:22:48 发布 656 收藏 11

分类专栏： 本科毕业设计总结 文章标签： 隐写分析

本文链接：https://blog.csdn.net/DRZ_2000/article/details/118599523

文章目录

一 图像隐写分析的数据集

小编本科毕设的题目是《基于深度学习的图像隐写分析》，**图像隐写分析问题本质上是一个二分类（binary classifier）问题**，即判断一张图片中是否含有隐秘信息，若包含隐秘信息则输出1，否则输出0。

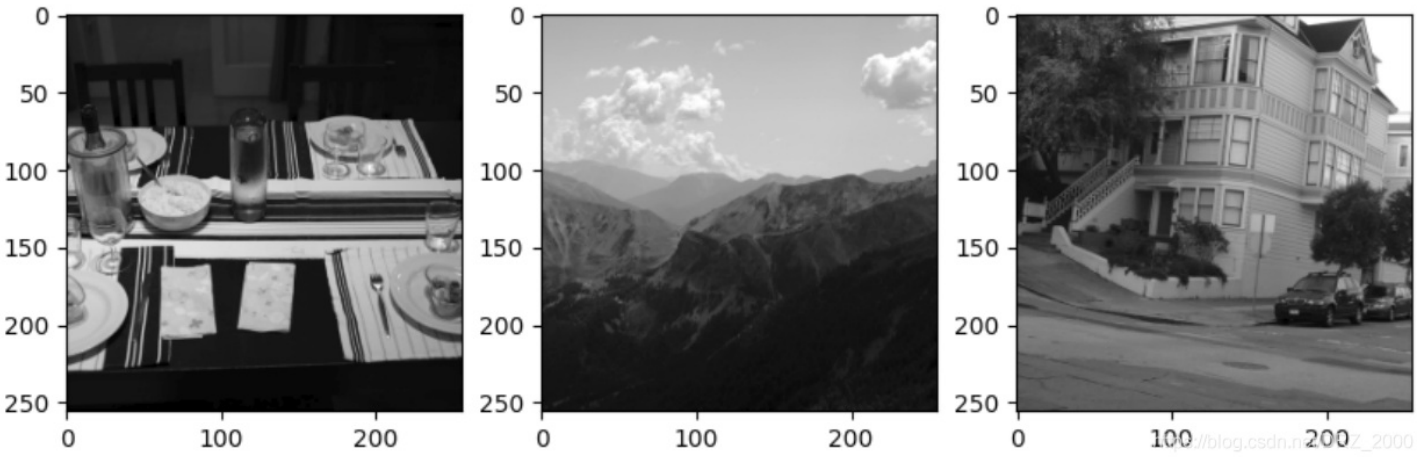
注意区分隐写术&隐写分析 (Steganography&Steganalysis) 和 数字水印 (Digital WaterMarking) 两种技术，两者都是向图像或视频等信息载体中嵌入内容并且对嵌入后的信息载体进行分析，只是侧重点不一样。隐写术和隐写分析更注重于嵌入信息的隐蔽性，即如何操作才能让嵌入信息很难被敌人发现，敌人面对手中截获的图片无法判断该图像是含密图像还是原始图像。数字水印更注重嵌入信息的鲁棒性，含水印信息的图片在传输过程中会受到诸如：旋转、裁剪、噪声模糊、对比度亮度的调整等等攻击手段，我们需要保证：面对受到攻击后的图像我们依旧可以很好的从图片中提取之前嵌入的水印信息，嵌入到图像中的水印信息有较强的鲁棒性。

在实际应用方面，隐写分析可以用于防止秘密信息的泄露，犯罪分子可以将秘密信息嵌入到原始图片中，由于生成的含密图像和不含隐秘信息的原始图像两者从肉眼上几乎没有区别，所以犯罪分子就可以堂而皇之的将隐秘信息通过图片泄露出去，因此对犯罪分子而言嵌入信息的隐秘性很重要；数字水印技术在作品版权领域有很大作用，以视频为例，视频制作方可以将己方信息嵌入到原始视频中（嵌入后的视频和原始视频没有区别，不影响观看），当市场上出现盗版视频时，制作方可以从该视频中提取之前嵌入的版权信息，作为对溥公堂时的证据。视频在传播途中可能会出现盗录、模糊、裁剪、旋转等攻击，因此嵌入信息的鲁棒性很重要，如果视频受到攻击后无法很好的提取水印信息那就糟糕了。

卡！停！扯的有些远了，我们还是回归主题：如何生成含密图像。

隐写分析使用的图片从宏观上可以分成来两大类：不含隐秘信息的原始图像和包含隐秘信息的含密图像，其中原始数据集我们采用BOSSBase数据集，该数据集是隐写分析领域公用的数据集，数据集中包含10000张大小为512x512、后缀为pgm的灰度图。BOSSBase数据集下载地址为: http://dde.binghamton.edu/download/ 页面左下角Image DataBase部分就有BOSSBase 1.01数据集下载链接。

数据集中部分图片展示如下(网络模型需要，故将图片由512x512缩放至256x256)：



获得原始图像后，我们接下来生成含密图像，这里我们直接使用现有的隐写算法，下载地址为：http://dde.binghamton.edu/download/stego_algorithms/ 由于本机为ubuntu系统，故仅选用了C++版的S-UNIWARD、HUGO、WOW三种图像空域隐写算法。

SPATIAL DOMAIN

Name	Matlab / MEX	C++ Win	C++ Linux	Proposed
S-UNIWARD *	S-UNIWARD.m	S-UNIWARD.zip	S-UNIWARD.tar.gz	[1]
WOW *	WOW.m	WOW.zip	WOW.tar.gz	[2]
HUGO bounding dist.	HUGO_bounding.m	HUGO_bounding.zip	HUGO_bounding.tar.gz	[3]
MG	MG.zip	-	-	[7]
Pentary MVG	MVG.zip	-	-	[8]
MiPOD	MiPOD.zip	-	-	[10]
Synch	Synch.zip	-	-	[9]

JPEG

Name	Matlab / MEX	C++ Win	C++ Linux	Proposed
J-UNIWARD *	J-UNIWARD.m	J-UNIWARD.zip	J-UNIWARD.tar.gz	[1]
nsF5	nsF5_matlab.m	-	-	[4] Details

Note: Some JPEG domain steganographic algorithms implemented in Matlab require ~~Phil Sallee's MATLAB Jpeg Toolbox~~ (jpeg_read routine). Update: the website is no longer active. With the author's permission, we provide the copy of relevant files (including a few pre-compiled verions) here.

下载隐写术后解压，以S-UNIWARD隐写术为例如下。

```
compile.sh  executable    images_stego  lib      S-UNIWARD_src
demodir     images_cover  include       matlab
```

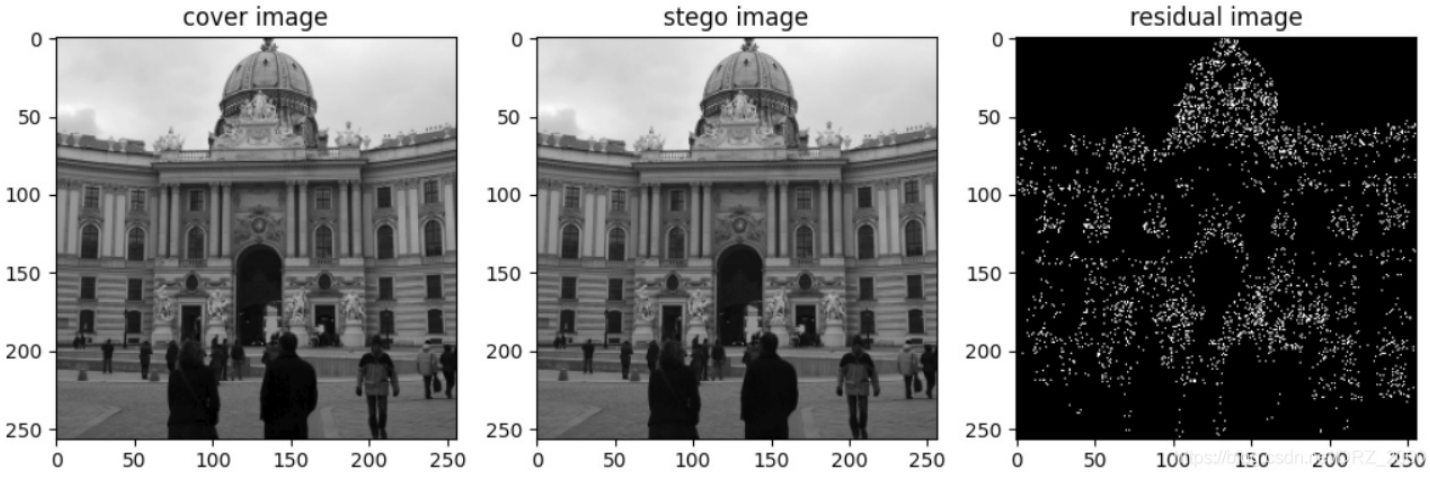
进入executable文件夹，`bash example_default.sh`运行脚本后，将images_cover文件夹下的原始图像嵌入秘密信息生成含密图像并保存至images_stego文件夹中。我们来看一下 example_default.sh脚本文件中的内容。

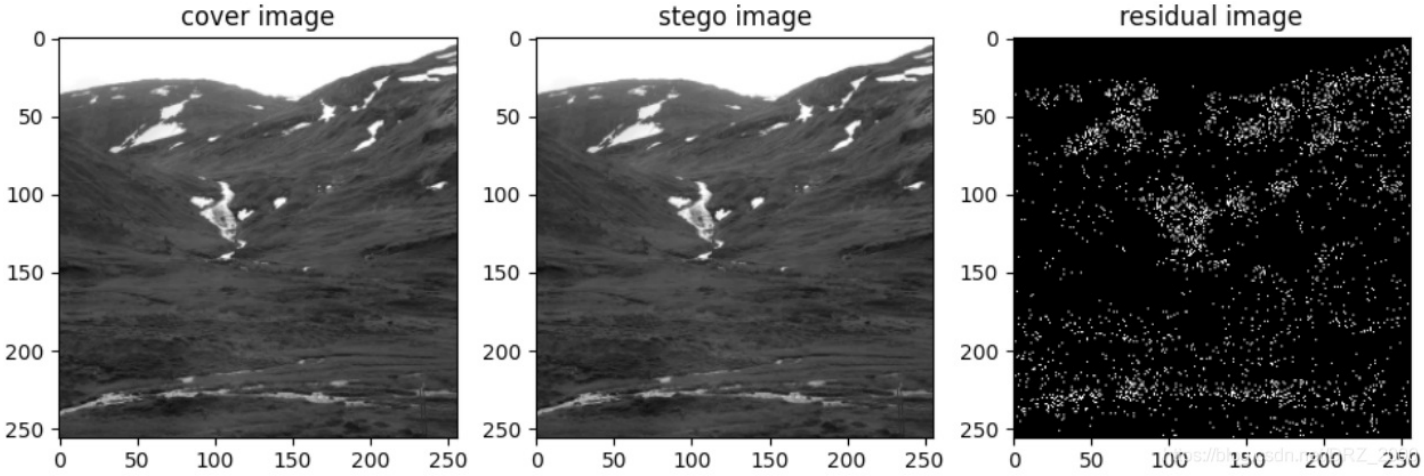
```
./S-UNIWARD -v -I ../images_cover -O ../images_stego -a 1
```

执行同目录下的S-UNIWARD可执行文件，-v表示输出嵌入过程中的过程信息、-I表示输入图片路径也就是原始图片所在文件夹的路径、-O表示输出路径也就是生成的含密图片保存的路径、-a表示嵌入率，此处嵌入率为1.0bpp (bit per pixel)。我们自己使用时，只要修改输入输出图片的路径以及相应的嵌入率即可。

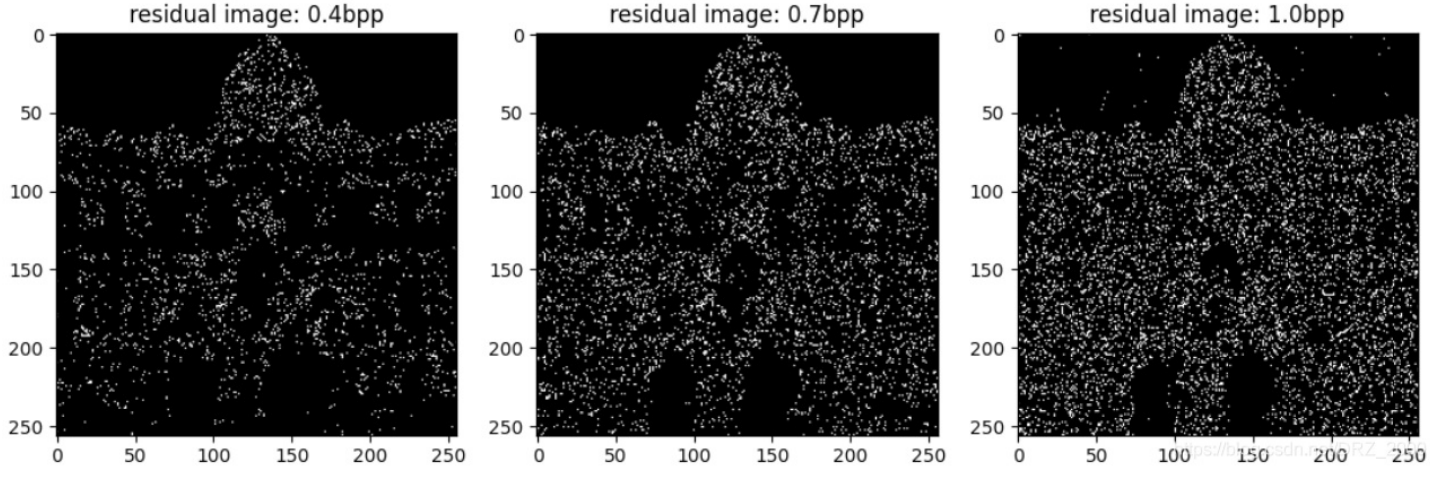
二 含密图像展示

本文采用的隐写术为：S-UNIWARD、HUGO、WOW，采用的嵌入率为：0.4bpp、0.7bpp、1.0bpp三种，我们查看生成的含密图像和原始图像以及两者之间的残差图像residual image，显示如下：





上图中从左到右分别为：原始图像、含密图像、残差图像。观察原始图像和含密图像发现，两者从肉眼上来看没有明显区别，视觉效果非常接近，这也从侧面说明隐写分析的难度。我们再来看一下不同嵌入率下得到的残差图像，显示如下：



上图中从左到右对应的嵌入率分别为0.4bpp、0.7bpp、1.0bpp。观察图像可以发现：嵌入率越大，残差图像越明显，即隐写术对原始图像的修改越大，隐写分析的难度也自然依次递减。计算不同嵌入率下含密图像的PSNR值分别为: 58.37、55.39、53.37。对于PSNR值我们通常认为PSNR值在40以上就说明图像和原始图像比较接近，而此处即使嵌入率最大为1.0bpp，得到的PSNR值也在50以上。

如果说之前肉眼观察原始图像和含密图像，两者十分接近是人的主观判断，那么此时PSNR值则是从实验数据方面证明隐写术对原始图片的修改很小，隐写分析的难度较大。

三 结语

10k张图片用于训练和测试，其实还是较少，还可以从ImageNet和COCO数据集中选用部分图片以做补充。本文从COCO数据集中另外选用了10k张图片，总图片数量共2万张，之后**按照14：1：5的比率划分为训练集、验证集和测试集**。至此，在数据集方面准备完毕，接下来要做的就是编写代码实现隐写分析模型。