YEDROUDJ-NET: AN EFFICIENT CNN FOR SPATIAL STEGANALYSIS【Yedroudj-Net:一个高效的空间隐写分析CNN】

原创

CV误会了我 于 2021-11-01 15:42:07 发布 98 收藏 1

文章标签： cnn r语言 深度学习

本文链接：https://blog.csdn.net/wangsanNOLOVE/article/details/121078378

YEDROUDJ-NET: AN EFFICIENT CNN FOR SPATIAL STEGANALYSIS

【Yedroudj-Net:一个高效的空间隐写分析CNN】

ABSTRACT

在大约10年的时间里，通过用Rich特征训练的集成分类器来检测隐藏在图像中的秘密信息。近年来，Xu等人的研究表明，设计良好的卷积神经网络(CNN)可以达到与两步机器学习方法相当的性能。在本文中，我们提出了一个在错误概率方面优于最先进的CNN。这个命题是对最近提出的命题的延续，是对不同论文中使用的重要砖块的巧妙融合。在CNN的基本部分中，可以引用预处理滤波器组和截断激活函数的使用，与Scale层关联的批归一化的五个卷积层，以及充分大小的完全连接部分的使用。一个扩充的数据库也被用于改进CNN的训练。我们用S-UNIWARD和WOW嵌入算法对CNN进行了实验评估，并将其性能与其他三种方法进行了比较:集成分类器加Rich模型和另外两种CNN隐写分析器。

1. INTRODUCTION

使用深度学习方法进行隐写分析的第一次尝试可以追溯到2014年的[1]自动编码器。一年后Qian[2]和Pibre 等[3]提出使用卷积神经网络。2016年，cnn[4]系统获得了与最先进技术相近的第一个结果。Xu-Net1[5] CNN被用作CNN集成的基础学习器。2017年还提出了其他网络，这次是用于JPEG隐写分析。在[6]中，作者提出了一种受Rich模型启发的预处理方法，并使用了一个大型学习数据库。结果接近于最先进的技术。在[7]中，网络是用受JPEG压缩过程启发的相位分割建立的。为了获得比最先进的结果稍好一点的结果，需要一组cnn。在[8]中，受ResNet[9]启发，使用了短连接技巧和20层的CNN也提高了结果的准确性。
这些结果非常令人鼓舞，但关于使用深度学习方法在其他图像处理任务中获得的收益[10]，与使用集成分类器[11]和丰富模型[12,13]或具有选择通道感知的丰富模型[14,15]的经典方法相比，隐写分析结果并没有“提高10%”. 2017年，改善CNN结果的主要趋势是：使用CNN集合，通过模仿丰富模型提取过程修改拓扑，或使用ResNet。在大多数情况下，设计或实验工作对性能的改善非常小。

通过回顾深度学习的良好实践以及最近的研究，我们实验设计了一种CNN用于空间隐写分析，其效率自然优于现有技术。执行此操作时，无需借助特定于图像性质的设计（空间、jpeg等）或CNN集成（已知可改善结果）。我们专注于CNN的设计，避免使用已知的技巧来提高性能，如转移学习[16]或虚拟扩充数据库[17]等。此外，提议的网络对超参数的初始化不敏感，因此容易收敛，这将在第3节稍后讨论。我们将该网络命名为“Yedurodj-Net”CNN，并将其与Xu-Net[5]、Ye-Net[17]以及与空间隐写分析用空间丰富模型[12]提供的集成分类器[11]进行比较。

2. YEDROUDJ-NET

图1展示了CNN的整体架构。该网络由预处理块、五个卷积块和由三个全连接层组成的全连接块组成，再加上一个softmax。网络在两个类别标签上生成一个概率分布。

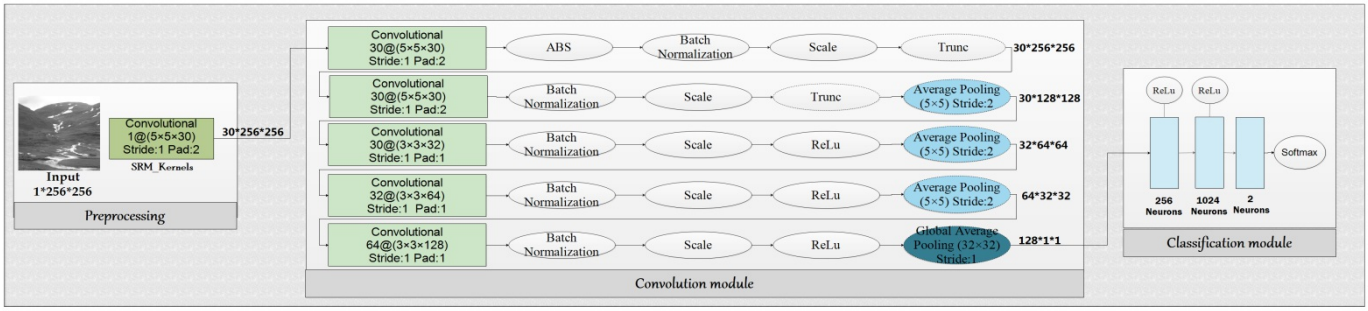

Fig. 1. Yedroudj-Net CNN architecture.

预处理块滤波器使用预定义的高通滤波器对输入的cover/stego图像进行预处理，以提取噪声成分残差。经过预处理的图像然后输入网络。以前的研究[2,3]发现，如果没有这个初步的高通滤波器，CNN的收敛速度会更慢。这种预处理极大地抑制了图像内容，缩小了动态范围，从而增加了弱隐写信号(如果存在)与图像信号之间的信噪比。因此，CNN可以在更紧凑、更稳健的信号上学习。

受到多样性[12]的启发，类似于[17]，我们使用来自SRM[12]的30个基本高通滤波器，而不是只使用一个滤波器，如[2,3,5]，来对输入图像进行预处理。注意，预处理块的过滤器内核值，即权值，在训练中并没有优化/学习。这种预处理已经被集成到一种懒惰的方式中，直接集成到CNN中，这样所有核的大小(权重矩阵)都被设置为5×5。它们的中心部分使用SRM内核的权值进行初始化，其余元素被填充为零。不执行内核值的规范化。

CNN的其他可分为卷积模块,用于特征表示,将输入图像转换成特征向量,并分类模块,组成的三个全层和一个softmax层,产生的分类决策(cover或stego)。

与Xu-Net类似，卷积模块有5个块，标记为“Block 1”到“Block 5”，用于提取有效的特征，用于cover和stego图像的识别;参见图1。每个区块由以下步骤组成:

1、一个卷积层。与Xu-Net[5]类似，对于区块1和2，我们将卷积内核的大小设置为5×5，但是对于区块3到5，我们将其减少为3×3。对于所有的卷积层，类似于Res-Net[9]和Xu-Net[5]，没有使用偏差。偏差项在卷积层设置为false，并移动到Scale层。

2、绝对值激活(ABS)层。这个ABS层只在Block 1中使用，类似于Xu-Net。它迫使统计建模考虑噪声残差的符号对称性。在Xu-Net[5]中观察到这一层的相关性。

3、批处理归一化(BN)。BN将每个特征的分布归一化为零均值和单位方差，并最终缩放和转换分布。使用BN层的好处在于，它可以降低训练对参数初始化的敏感性[18]，允许使用更大的学习速率，从而加快学习速度，并提高检测精度[7]。注意，与ResNet[9]类似，与Xu-Net相反，我们提供了一个BN层和一个缩放层。后者试图更有效地学习缩放和转换参数。这两个参数可以通过独立的Scale层很好地学习。与ResNet类似，我们观察到网络的准确性有很小的提高。

4、非线性激活层。对于block 1和block 2，使用截断函数来限制数据值的范围，并防止更深的层对大的值建模。事实上，这些值是稀疏的，没有统计学意义。截断函数(Trunc)的公式如Eq. 1所示，其参数化为t∈N，一个阈值:

$$Trunc(x) = \begin{cases} -T, & x < -T, \\ x, & -T \le x \le T, \\ T, & x > T. \end{cases} \qquad (1)$$

这个在[17]中提出的异常值抑制过程，也可以看作是使用了鲁棒性函数。对于3到5块，由于其性能好，梯度计算速度快，所以使用了经典的整流线性单元(ReLU)。

5、平均池。这个平均池化层只在块2到5中使用。这允许对特征图进行取样，从而降低维度。对于最后一个块，进行全局平均池化，为每个对应的feature map逐个生成一个元素，从而防止统计建模从训练数据[19]中获取嵌入像素的位置信息。第一个块没有池，避免了网络开始时的信息丢失。

从卷积模块中提取的特征提供给分类模块，该模块由三个完全连接的层组成。第一层和第二层神经元数量分别为256和1024，最后一层全连接层只有2个神经元，对应网络输出的类数。在本模块的最后，使用softmax激活函数生成两个类标签上的分布。

3. EXPERIMENTS

3.1. Dataset and software platform

我们使用了S-UNIW ARD[20]和WOW[21]，这两种著名的用于空间域内嵌入的内容自适应方法，以及它们的Matlab实现(在线代码2)，以及用于嵌入的模拟器和每个嵌入的随机密钥。因此，我们避免了c++代码的任何错误使用，如[3]中所报告的固定且唯一的嵌入键。

Our steganalysis CNN,Yedroudj-Net, is compared with the state-of-the-art approaches:Xu-NetCNN [5],Ye-NetCNN [17], and with SRM + ECwhich stands for the hand-crafted feature set Spatial-Rich-Model [12] and the Ensemble Classifier [11]. For a fair comparison, all the involved steganalysis methods are tested on the same subsampled images from the BOSSBase database v.1.01 [22]. All CNNs experiments were performed with the publicly available Caffetoolbox [23] with necessary modifications, plus digits V5. All tests were run on an NVidia Titan X GPU card.

3.2. Training, Validation, Test

由于我们的GPU计算平台和时间限制，我们所有的实验都是在256×256像素的图像上进行的，类似于[17]。为此，我们使用带有默认参数的imresize()Matlab函数将所有512×512图像重新采样到256×256图像。然后，我们的256×256 BOSSBase被分成两组，50%(分别是50%和50%)。另外50%的cover/stego对被分配到训练中。测试)。从5000对训练组合中随机选择4000对用于训练，其余1000对用于验证。在训练阶段，测试集保持不变。

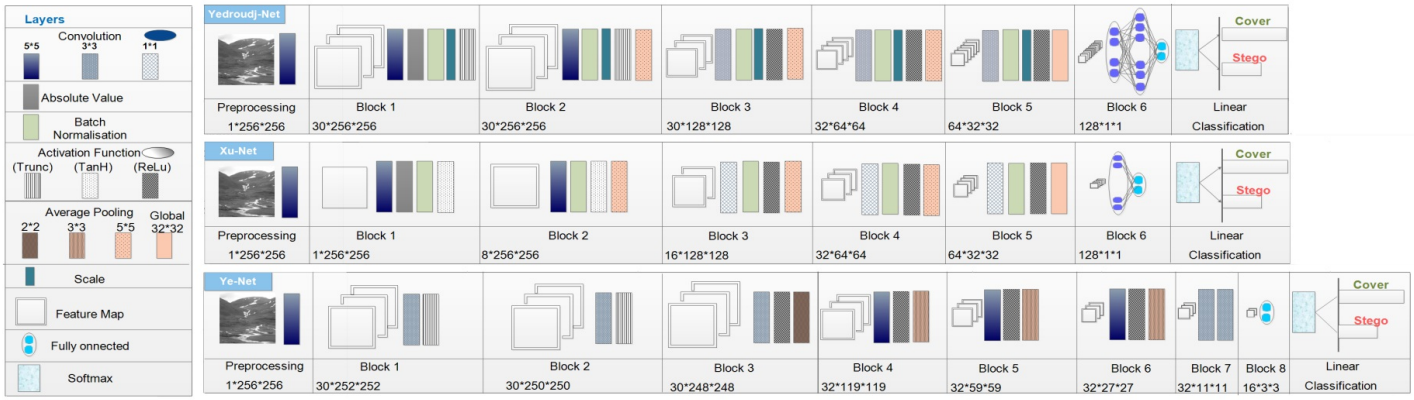

Fig. 2. Comparison of Yedroudj-Net, Xu-Net, and Ye-Net architectures.

在cnn训练期间，我们最多固定了900个epoch。然而，大多数情况下，当出现过拟合现象时(通常在WOW的epoch 200和S-UNIWARD的epoch 300之前)，即当loss在训练集中继续减少，但在验证集中开始增加时，我们手动停止训练。在实践中，通过观察验证测试中计算出的损失曲线，我们保留了两个版本的CNN:损失最小的CNN模型(分别是损失最小的CNN模型)。最大值)。在测试集上对这两个CNN模型进行评估，我们报告这两个CNN模型检测的平均错误概率。

对于SRM + EC，我们使用尺寸为34 671[12]的SRM特征集，以及集成分类器[11]。我们报告在相等先验下的最小误差概率，平均超过10次测试。

3.3. Hyper-parameters

我们使用小批量随机梯度下降(SGD)来训练我们的CNN。动量固定为0.95，重量衰减为0.0001。不使用dropout。由于GPU内存限制(8对覆盖/隐写对)，训练过程中的批大小被设置为16。所有层初始化使用xavier方法:权值遵循高斯分布和选择，以便每个层之间的输入和输出的方差保持相同[24]。在培训过程中，我们使用Caffe的step策略来调整学习率(初始化为0.01)。有了这个策略，总纪元数的每10%，我们的学习率就减少一个等于0.1的因子。截断函数(见式1)的阈值T为第一层的3，第二层的2,30个基本高通SRM滤波器未归一化。

3.4. Difference between the 3 CNNs

在本节中，我们将简要讨论我们的CNN yedroudjnet、Xu-Net CNN和Ye-Net CNN之间的区别，它们是用于空间隐写分析的最先进的CNN。在我们的比较中，Xu-Netis是一个类似于[5]中给出的CNN，它拍摄的图像大小为256×256而不是512×512。因此，我们抑制了第一个块的平均池，这是一个有利的措施，因为它避免了早期的降采样。我们还在全连接层之间设置了一个ReLU激活函数。图2显示了所有cnn的总体架构。下面我们总结一下cnn之间的主要相似点和不同点:

1、Yedroudj-Net和Xu-Net都使用5个卷积层。然而，Yedroudj-Net在全连接部分的输入部分有两倍多的特征(256)。Ye-Net有更多的卷积层。

2、Yedroudj-Net和Xu-Net都使用批处理归一化层;而Ye-Net则不然。

3、Yedroudj-Net和Xu-Net都使用了绝对值层（ABS）;而Ye-Net则不然。

4、Yedroudj-Net和Ye-Net都使用30滤波器组进行预处理;Xu-Net则不然。

5、Yedroudj-Net和Ye-Net都在Block 1和Block 2中使用截断激活函数(我们“实验”发现，仅在Block 1和Block 2中使用截断激活函数在检测精度方面是最好的选择，这些实验在这里没有报道);Xu-Net则不然。

6、Yedroudj-Net有三个(分别是:Xu-Net两层，Ye-Net一层)全连通层。

3.5. Results without using any tricks

3.5.1. General performance comparisons

在表1中，我们报告了在0.2 bpp和0.4 bpp下隐写WOW和S-UNIW ARD嵌入算法时得到的错误概率。隐写分析方法有Yedroudj-Net、Xu-Net、Ye-Net和SRM+EC[11,12]。

Table 1. Steganalysis error probability comparison of Yedroudj-Net, Xu-Net, Ye-Net, and SRM+EC for two embedding algorithms WOW and S-UNIWARD at 0.2 bpp and 0.4 bpp.

	BOSS 256×256			
	WOW [21]		**S-UNIWARD** [20]	
Payload / **Steganalysis**	0.2 bpp	0.4 bpp	0.2 bpp	0.4 bpp
SRM+EC [11, 12]	36.5 %	25.5 %	**36.6** %	24.7 %
Yedroudj-Net	**27.8** %	**14.1** %	**36.7** %	**22.8** %
Xu-Net [5]	32.4 %	20.7 %	39.1 %	27.2 %
Ye-Net [17]	33.1 %	23.2 %	40.0 %	31.2 %

与其他CNN算法相比，我们所提出的CNN算法取得了远远优于其他算法的结果。Yedroudj-Net在两种嵌入算法和两种有效载荷方面比Xu-Net好2%到6%。与Ye-Net相比，这一结果甚至更好，Yedroudj-Net比Ye-Net高出3%至9%。让我们注意到，当与SRM+EC相比时，其他两个cnn并不总是更好的。为了击败SRM+EC，这些方法需要使用CNN的集成，如[4]中提出的，或增加学习数据库，如[6]中提出的，如下节所示。

请注意，必须非常谨慎地初始化Ye-Net的学习率和管理其在各个epoch的演变。实际上，错误的初始化会阻止网络聚合。在Yedroudj-Net和Xu-Net中，批处理归一化的使用确保了对此类参数设置的较小灵敏度。

在这些一般比较的基础上得出结论，在没有任何通道感知的经典透视场景中，并且不使用集合、更大的数据库、数据库的虚拟扩充或迁移学习，Yedroudj-Net比所有最先进的方法都具有明显的优势。

3.6. Results with a Base augmentation

改进CNN的结果存在许多技巧，但为了更好地利用深度学习方法的能力，基础增强似乎是一个非常重要的措施。

Table 2. Base Augmentation influence: error probability comparison of Yedroudj, Xu and Ye nets on WOW at 0.2 bpp with a learning base augmented with BOWS2, and Virtually Augmented.

	BOSS	BOSS+BOWS2	BOSS+BOWS2+VA
Yedroudj-Net	**27.8** %	**23.7** %	**20.8** %
Ye-Net	33.1 %	26.1 %	22.2 %
Xu-Net	32.4 %	30.3 %	30.5 %

在机器学习中，这对cnn也是如此，使用足够大的训练基地来确保良好的泛化，但也要避免过度训练。一些作者倾向于使用大型数据库[2,6,17]，以获得最先进的结果。在上述实验中，我们试图研究在不修改测试集的情况下增加学习数据库的大小所带来的改善。这意味着学习集不只是包含与测试集相同类型的图像:例如，摄像机的设置，学习集的场景，都可以与测试集的不同。我们在表2中展示了增加图像数据库对错误概率的影响。为了增加训练集的大小，我们根据[17]测试了两个场景。

在第一个场景中，BOSS + BOWS2指出，我们将有效负载嵌入了二次抽样的BOSSBase数据库v中。1.01 [22]. 我们将这个基础分成两组：50%的cover/stego对用于训练集，其余用于测试集。然后，将10000对额外的cover/stego对（通过对BOWS2Base[25]进行二次抽样获得）添加到训练集中。学习数据库现在包含15000对封面/隐秘图像，减去BOSS提供的1000对，用于验证。

在第一个场景中，BOSS + BOWS2+VA指出，我们将有效负载嵌入了二次抽样的BOSSBase数据库v中。1.01 [22]. 我们将这个基础分成两组：50%的cover/stego对用于训练集，其余用于测试集。然后，将10000对额外的cover/stego对（通过对BOWS2Base[25]进行二次抽样获得）添加到训练集中。学习数据库现在包含15000对cover/stego图像，减去BOSS提供的1000对，用于验证。

表2显示了Yedroudj-Net、Xu-Net[5]、Ye-Net[17]与负载为0.2 bpp的嵌入算法WOW[21]的检测误差率性能比较。对于所有算法，使用BOSS+BOWS2比只使用BOSSBase获得了更好的性能。YedroudjNet得到了最好的结果，其检测错误概率降低了4%。Ye-Net和Xu-Net的检测误差率分别降低了7%和2%。在这一点上，尚不清楚改善只是由于缺乏数据，还是因为额外的图像来自相同的相机。尽管如此，我们还是进行了额外的实验，在[26]论文中报道，似乎为了提高性能，必须增加来自相同来源的图像的数据库，并根据像素分辨率和比率进行开发。

当虚拟地增加整个BOSS+BOWS2学习集(即BOSS+BOWS2+VA)时，由于8个不引入插值的旋转和翻转组合，性能再次提高。与仅使用BOSSBase进行训练的情况相比，Yedroudj-Net保持了最好的结果，并将检测错误概率降低了7% (Ye-Net降低了11%，Xu-Net降低了2%)。与RM+EC[11, 12]相比，在BOSSBase上学习时，RM+EC的误差率为36.5%，Yedroudj-Net的误差率为20.8%，提高了16%。Ye-Net提高了14%，旭网提高了6%。

这些测试表明，当使用5-7块CNN时，拥有一个大数据库是多么重要。参数的数量(不考虑BN和/或尺度)大约从5万(Xu-Net)到50万(Yedroudj-Net)。如此大量的未知需要有足够的样本。实验表明，cnn仍然缺乏足够的学习样本。对于5-7块的CNN基于BOSSBase隐写分析，即使是112,000对(BOSS+BOWS2虚拟增强)图像也不够。因此，即使收敛时间增加，使用更大的基数也可以使我们的CNN获得更好的性能。

在32G内存的Intel Core i7-5930K CPU 3.50GHz×12上使用上一代GPU卡(Nvidia TitanX)，在bosbase上学习yedroudj.net CNN不到一天，在BOSS+BOWS2上学习3天，在BOSS+BOWS2+VA上学习7天以上。

4. CONCLUSION

本文介绍了用于空间隐写分析的Yedrodj Net CNN的评估。本CNN收集了一些最新的设计主张，以便在不了解选择频道的情况下，在经典透视场景中构建一种超越最先进方法的简单方法。

隐写分析性能改进的关键是以下元素的组合：用于预处理步骤的一组过滤器、截断激活函数和与缩放层相关联的批标准化。

另外一个处理学习库大小问题的实验表明，通过添加BOWS2和虚拟地增加学习库，结果非常令人满意。在0.2 bpp下的WOW实验中，与RM+EC相比，误差概率降低了16%。