变换域隐写术检测分析

转载

c2a2o2 于 2020-05-25 08:57:49 发布
原文链接：https://www.anquanke.com/post/id/193391

前言

犯罪分子经常使用隐写术将秘密指令隐藏至图片中做非法操作，以躲避专业人员的检测，因此需要隐写术检测分析来阻止这种恶意行为。本文提出了一个变换域的隐写分析模型，以检测图像中是否使用变换域隐写算法隐写秘密指令，它具备良好的检测效果。

背景

隐写分析是通过对载体的统计特性进行分析，判断载体中是否隐藏有额外的信息的技术。目前的隐写分析研究领域通常将隐写分析看成一个二分类问题，目标是区分正常载体和含密载体。在这种情况下，现有的方法主要通过以下两个步骤来构建隐写分析检测器：特征提取和分类。

在特征提取步骤中，一系列手工设计的特征会从图片中被提取出来，以捕获嵌入操作的影响。隐写分析效果的好坏很大程度上依赖于特征设计。然而，由于缺乏精确的自然图像模型，这个工作变得十分复杂。目前最可靠的特征设计方式是首先计算噪声残差，然后利用相邻元素的条件或联合概率分布对残差进行建模。随着隐写术越来越复杂，隐写分析领域特征设计过程中需要考虑的图像统计特性也更复杂，这进一步加剧的手工设计特征的难度。实际上，特征也逐渐朝着复杂化、高维化发展。例如代表性的隐写分析方法SRM (Spatial Rich Model),PSRM (Projection Spection Rich Model)等特征维度均超过了10,000 维。在分类步骤中，SVM或集成分类器等分类器会学习提取出来的特征并用于分类。特征提取和分类步骤是分离的，它们不能进行统一的优化，这意味着分类器可能无法充分利用特征提取中的有用信息。

为了解决以上问题，学者们将深度学习的理论引入隐写分析中。根据隐写信息提取的域，图像隐写分析可以被分为两大类：空域隐写分析和变换域隐写分析。针对于空域隐写分析的研究成果较多，而针对变换域隐写分析的研究成果较少。2016年Zeng等人第一次将CNN应用于变换域隐写分析，提出了一个含有三个CNN子网络的JPEG隐写分析模型，实验结果超过了传统的DCTR和PHARM，这是很大的进步，但网络结构复杂参数量大。

变换域隐写分析模型

本文提出了一个隐写分析模型，主要从预处理模块、特征提取模块和分类模型三部分介绍它。

预处理模块

预处理模块通常由一个高通滤波器(HPF)层组成。HPF层是一种特殊的卷积层，处于整个网络的最前端。在这一层中，通常会使用预定义的HPF进行隐写特征过滤操作。本模型的HPF来自于空域富模型(SRM)，大小为3×3，并且进行了归一化，使得滤波器中的权重绝对值分布在0、1之间。如下图所示。

$$\frac{1}{4}\begin{pmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{pmatrix}$$

使用空域富模型SRM的HPF的原因如下：首先，使用通过空域HPF获取的像素残差而不是量化的离散余弦变换（DCT）系数，更利于隐写分析。隐写算法对变换域的JPEG量化系数的修改幅度较小，因此对JPEG量化系数的统计特征影响并不显著，则使用DCT系数从变换域学习样本数据的差异没有明显优势。对空域而言，情况则完全不同。虽然量化系数只改变了1个单位，但经量化步长的放大，隐写操作对空域像素的干扰会被进一步放大，有利于隐写分析。此外，SRM已经给出邻域像素相关性模型，在空域分析有明确的模型作为指导，像素残差抑制了图像内容的干扰，通过邻域像素值相关性更好地表现了隐写特征。然而邻域像素相关性在变换域的表达没有良好的建模，对变换域的DCT系数的分析将不可避免地受到图像内容本身的影响，这进一步表明在空域对残差进行分析的合理性。第二，合理大小的滤波核更有利于隐写分析。研究指出，对于复杂图像来说，邻域像素相关性会随着边界与中心的距离的增大而剧烈下降。因此，本模型选用3×3大小的滤波器。

特征提取模块

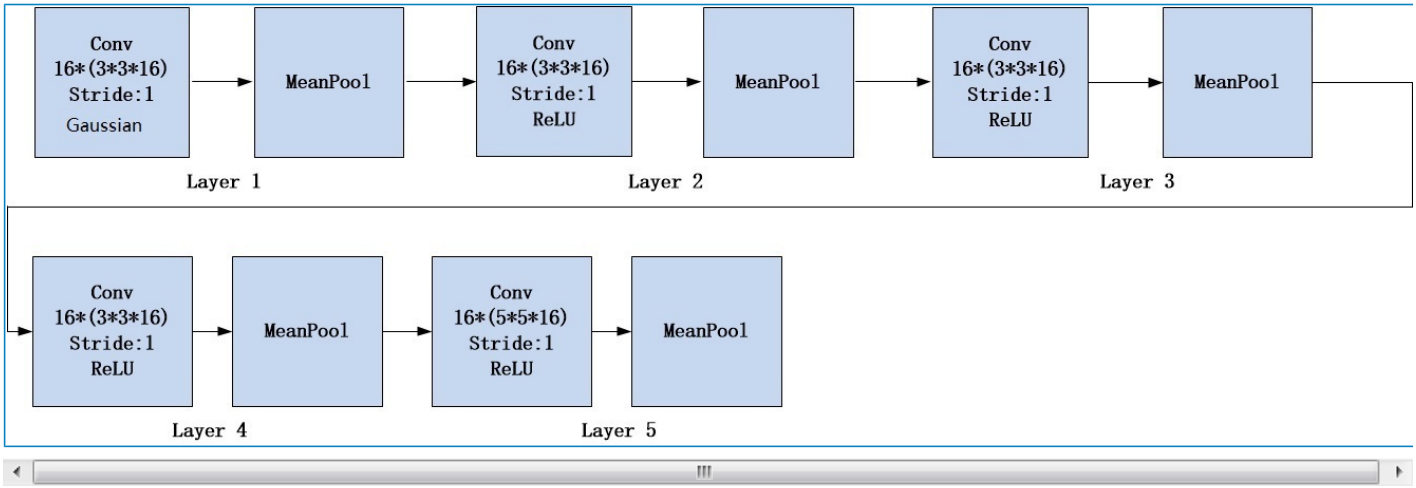
图片经过预处理模块后，提取出的隐写特征残差会被输入到由数个卷积层组成的特征提取模块中。

特征提取模块通常由多个卷积层组成。卷积层的输入和输出是一组称为特征图的数组，而每个卷积层通过三个步骤生成特征图，分别是卷积、非线性激活和池化。在实践中，第一步先使用k个卷积核进行过滤，从而生成k个新的特征图。使用F^{n}(X)表示第n层输出的特征图，W^{n}表示第n层的卷积核，B^{n}表示偏置，则卷积层可以表示为：

$$F^n(\mathrm{X}) = \mathrm{pooling}\left(f^n\left(F^{n-1}(\mathrm{X}) * W^n + B^n\right)\right)$$

其中F^{0}(X) = X代表输入数据，f^{n}(·)是非线性激活函数。非线性激活函数会应用于每个输入的元素之上，典型的激活函数有sigmod、TanH和ReLU等。pooling(·)代表池化操作，包括平均池化和最大值池化。通常来说非线性激活函数和池化操作在特定的层中是可选的。

对于卷积层的运算，每个输出的特征图通过卷积结合多个输入的特征图的特征。卷积层的结构涉及到局部感知和权值共享的概念。对于局部感知来说，每一个低维度的特征只会从输入的一个子集中计算得出，比如一个图像中给定位置的像素的区域。这种局部特征提取器在应用于不同的相邻输入位置时共享相同的权重参数，这相当于图像像素值与包含权重参数的内核的卷积。权值共享产生了一个移位不变性的运算，同时也减少了自由变量的个数，从而增加了网络的泛化能力。特征提取模块的结构如下图所示。



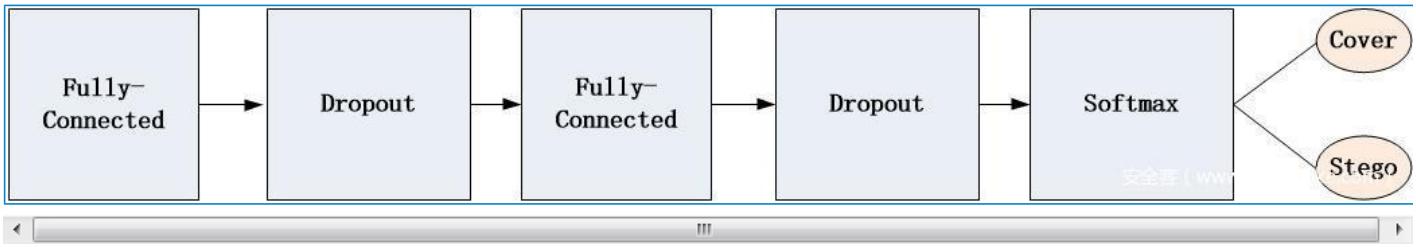
本模型的特征提取模块一共包括5层，每一层均包含16个大小为3×3或者5×5的卷积核。相比于已提出的方案，卷积层的个数大大地减少了。特征提取模块的第一层卷积层使用高斯非线性激活函数（Gaussian），后几层使用ReLU作为非线性激活函数。已知在CNN较深的层次使用ReLU作为激活函数更有利于特征的表达。在每个卷积层的最后使用了平均池化操作。池化操作的目的是将低层次的特征标识转换成更有用的特征标识，从而保存重要的信息，并丢弃无关的细节。一般来说，更深层次的特征表示需要来自逐步扩大的输入区域的信息。池化的作用是将信息合并到一组小的局部区域中，同时减少计算开销，这与量化、截断汇聚有用特征，抛弃无用信息的目的相似。

分类模块

特征提取模块提取的特征被输入到分类模块进行分类，分类模块最终会输出一个标签标明图片的所属类别。分类模块中通常含有几个全连接层以及一个分类器。

全连接层通常会包含较多的可训练参数，当训练集不够大时容易导致过度拟合。解决办法是使用dropout对全连接层进行正则化。当使用了dropout进行训练时，对应图层中的部分神经元输出以一定概率被置为0，这在一定程度上可以提高CNN的泛化能力。

本模型的分类模块的结构如下图所示。



其中，两个全连接层均包含128维的特征向量，加入的Dropout的参数为0.5。

模型实现与分析

网络的实现

在预处理层中，卷积核使用归一化的3×3高通滤波器进行了初始化，使用Keras框架，实现如下。

```
kernel = np.array([[[[ -1]],[[  2]],[[ -1]]],
                   [[[  2]],[[ -4]],[[  2]]],
                   [[[ -1]],[[  2]],[[ -1]]]], dtype=np.float)
kernel /= 4
bias = np.array([0])

model = Sequential()#特征提取模块
model.add(Conv2D(1, kernel_size=(3, 3),input_shape=input_shape, kernel_initializer=kernel, trainable=False))
```

特征提取模块和分类模块的实现代码如下。

```
model.add(Conv2D(16, (3, 3), activation='custom_activation',kernel_initializer='glorot_normal')) #卷积、非线性激活、池化
model.add(AveragePooling2D(pool_size=(3, 3),strides=2))                                          #124
model.add(Conv2D(16, (3, 3),padding='same',activation='relu',kernel_initializer='glorot_normal'))
model.add(AveragePooling2D(pool_size=(3, 3),strides=2))                                          #16x61x61
model.add(Conv2D(16, (3, 3), activation='relu',kernel_initializer='glorot_normal'))
model.add(AveragePooling2D(pool_size=(3, 3),strides=2))                                          #16x29x29
model.add(Conv2D(16, (3, 3), activation='relu',kernel_initializer='glorot_normal'))
model.add(AveragePooling2D(pool_size=(3, 3),strides=2))                                          #16x13x13
model.add(Conv2D(16, (5, 5), activation='relu',kernel_initializer='glorot_normal'))
model.add(AveragePooling2D(pool_size=(3, 3),strides=2))                                          #16x4x4
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))                  #分类模块
```

数据集+参数

本文使用的数据集是标准化数据集BOSSBase 1.01，BOSSBase 1.01包含了10000张512×512大小的载体图片。由于计算资源的限制，在实验中使用MATLAB的imresize()函数将BOSSBase 1.01的图片缩放为256×256的大小，使用3种变换域的隐写术来评估模型的隐写分析效果，它们分别是：nsf5、MB1以及J-UNIWARD。

在实验中，使用隐写术和载体生成对应的10000张载密图片，这10000张载体图片和10000张载密图片共同组成一次实验的数据（10000对载体-载密图片）。训练集、验证集、测试集使用8:1:1的比率，即训练集为8000对载体-载密图片，验证集为1000对载体-载密图片，测试集为1000对载体-载密图片。

设置学习率为0.001，使用mini-batch随机梯度下降算法，mini-batch的大小为64。在预处理模块中，卷积核使用高通滤波器初始化，但被设置为不可训练。在特征提取模块中，所有卷积层的卷积核皆使用Xavier初始化器进行初始化，所有卷积层的池化大小为3×3，步长为2。

结果分析

分别对变换域隐写术nsf5、MB1以及J-UNIWARD做隐写分析，嵌入隐写信息的载荷分别为0.2、0.5，结果如下。

算法	载荷Payload	准确率
nsf5	0.2	0.7355
	0.5	0.9230
MB1	0.2	0.9492
	0.5	0.9750
J-UNIWARD	0.2	0.7060
	0.5	0.8485

可见，模型对于nsf5和MB1的隐写检测效果显著，尤其对于MB1，在payload为0.2时检测的准确率为0.9492。然而对于J-UNIWARD的隐写检测效果不显著，认为这是由于没有很好地识别和定位此算法嵌入隐写信息的位置造成的。

综上，本模型的的参数规模的计算量较小，且具有良好的隐写检测效果，未来可研究分析J-UNIWARD的隐写算法机理改进隐写分析模型，使其具有更优的性能。

模型核心代码链接：

https://github.com/HYWZ36/Steganalysis-in-frequency-domain