Detection of Diversified Stego Sources with CNNs

Jan Butora and Jessica Fridrich, Department of ECE, SUNY Binghamton, NY, USA, {jbutora1, fridrich}@binghamton.edu

Abstract

The goal of this article is construction of steganalyzers capable of detecting a variety of embedding algorithms and possibly identifying the steganographic method. Since deep learning today can achieve markedly better performance than other machine learning tools, our detectors are deep residual convolutional neural networks. We explore binary classifiers trained as cover versus all stego, multi-class detectors, and bucket detectors in a feature space obtained as a concatenation of features extracted by networks trained on individual stego algorithms. The accuracy of the detector to identify steganography is compared with dedicated detectors trained for a specific embedding algorithm. While the loss of detection accuracy w.r.t. increasing number of steganographic algorithms increases only slightly as long as the embedding schemes are known, the ability of the detector to generalize to previously unseen steganography remains a challenging task.

Introduction

The objective of steganography is to communicate secretly in a covert manner, meaning the very existence of the secret exchange cannot be established by observing the communication channel. Steganalysis, on the other hand, deals with detecting the use of steganography. The secret message is typically embedded by slightly modifying the individual elements of the cover object. For covers in the form of images in raster formats, the embedding modifications are often restricted to changes by ±1. This makes detection of steganography a very challenging problem. In the vast majority of research papers today, the steganalyst adopts the Kerckhoffs' principle and assumes that the source of cover images, the embedding scheme, and the size of the secret payload are known. And even within this unrealistically sandboxed environment, reliable detection of modern content-adaptive embedding schemes in a single image is not possible if the steganographers restrict the size of their secret payload [3, 40].

In this paper, we take a look at the problem of detecting diversified stego sources to address the situation in which the steganalyst does not know exactly which steganographic scheme was used by the steganographer. We consider both the closed-set problem in which all stego algorithms that can be potentially used are known and the much more challenging open-set setup when the embedding algorithm may not be known to the steganalyst. We work with detectors built as convolutional neural networks (CNNs) for several reasons. First, they have been shown to provide significantly better performance than the outgoing paradigm – rich models with simple classifiers. Second, rich models are too "rigid" to adapt well to more complex stego sources. In contrast, CNNs jointly learn their filters as well as the classification. Finally, CNNs can be easily scaled up if necessary to contain the increased complexity of diversified stego sources, for example by increasing the number of filters or layers. In particular, we work with the recently proposed Steganalysis Residual Network (SRNet) [3] because it provides state-of-the-art performance among other competing network architectures [38, 37, 40, 4, 41, 15].

We limit our study to spatial domain steganography and investigate three different approaches: a binary one-against-all detector, a mutli-class detector, and a "bucket detector" built from binary detectors dedicated to detecting each steganographic method by fusing them on the "feature level." The most promising approach of the three is the multi-class detector, which is contrasted with previous art based on rich media models.

Relevant prior art

The problem of detecting multiple JPEG stego algorithms has been addressed by Pevný et al. in [25, 27, 26, 28]. The authors used detectors built using machine learning and low-dimensional features, and focused on the early steganographic schemes for the JPEG format: Jsteg [35], OutGuess [30], Steghide [11], JP Hide&Seek , F5 [36], and model-based steganography [32]. Special attention was paid to resolving missed detection and large false alarms due to double JPEG compression associated with F5 and OutGuess.

Cogranne et al. [5] proposed a multi-class detector with novel optimality criteria by leveraging the observation that the projections of feature vectors on the weight vectors of individual base learners in the FLD ensemble [19] can be well modeled as a multi-variate Gaussian (MVG) distribution. Under the assumption of the so-called "shift hypothesis" [16], embedding only impacts the mean of the MVG distribution but not its covariance, which allowed the authors to derive optimal minimax test that guarantees a prescribed false-alarm probability and maximizes the worst correct classification probability across all stego algorithms (all alternative hypotheses).

The important topic of building a universal steganalyzer capable of detecting an arbitrary steganographic scheme was investigated in [29]. In particular, the authors studied a multi-class detector trained to detect K steganographic algorithms using the max-wins strategy [28] by collecting votes from $\binom{K+1}{2}$ binary classifiers built between every pair of classes and implemented as Gaussian SVMs. Since this detector failed on the so-called –F5 algorithm (the F5 algorithm with the embedding operation with reversed polarity) which is otherwise easily detectable, the authors looked into the family of one-class detectors, namely one-class SVM, one-class neighbor machine [22], and density-level detection [34]. While such detectors better generalize to unknown embedding schemes (they are more universal), they are less reliable on known steganography.

A qualitatively new approach to identifying steganographic content has been proposed in [17]. There, the authors moved away from the problem of identifying individual stego images to the problem of identifying the steganographer by jointly considering multiple images sent by the same individual (pooled steganalysis [16]). Finally, we remark that the subject of detecting stego sources with unknown (diverse) payload has been studied in [23].

Steganalysis with CNNs

In this paper, we revisit the problem of detecting diversified stego sources armed with a novel machine learning tool – convolutional neural networks. The first detector of this type appeared in 2015 [31] with all subsequent network architectures [38, 37, 40, 4, 41, 15] retaining certain critical elements from the previous detection paradigm, the spatial rich model (SRM) [7], in the sense that the convolutional filters in the first layer were either fixed or initialized with heuristic values, such as SRM filters [40] or DCT kernels [37, 39], and the feature maps were thresholded, quantized [37, 39, 40, 41], or split by their JPEG phase [13, 4].

The first CNN free of the above heuristic elements in which all network parameters are learned in an end-to-end manner from randomly initialized values is the SRNet [3]. It is also the first architecture that is universal as it provides state-of-the-art performance for both spatial and JPEG domain. SRNet makes use of residual layers [9, 10] that prevent the vanishing gradient phenomena, allow the use of a deeper architecture, and encourage feature reuse in the training process. The SRNet assumes that the input image is a 256×256 grayscale tile.[1] All convolutional layers employ 3×3 kernels, batch normalization, and all non-linear activation functions are ReLU. The first eight convolutional layers use unpooled feature maps on their input because pooling can be detrimental for steganalysis as it reinforces content and suppresses the noise-like stego signal by averaging adjacent embedding changes. The first eight layers can thus be loosely viewed as noise residual extractors. Pooling in the form of 3×3 averaging, stride 2, is applied on the output of Layers 8–11. All 512 16×16 feature maps outputted by the last convolutional layer are globally pooled to form a 512-dimensional vector input into the Inner-Product (IP) layer, the classifier part of the network.

In this paper, we do not use the version of the SRNet aware of the selection channel (Section V in [3]) because we aim to detect a wide spectrum of both content-adaptive and non-adaptive steganographic schemes. We also note that SRNet can be easily modified to accept color images (three channels) by changing all 3×3 filters in the first layer with $3 \times 3 \times 3$ filters.

Extension to diversified stego sources

Here, we introduce three methods for steganalysis of diverse stego sources with CNNs that will be investigated in the following sections. Vectors and matrices will be typeset in boldface, and we reserve the calligraphic font for sets. The symbol K stands for the number of stego algorithms in our stego source. For two probability mass functions (pmfs), p, q, their cross-entropy and KL-divergence is $H(p,q)$ and $D_{\mathrm{KL}}(p||q)$, $H(p)$ stands for the entropy of p.

Cover images will be denoted with c, stego methods S_k will be indexed with $k \in \{0, \ldots, K\}$, where $k = 0$ is reserved for the cover class. The stego version of cover c is obtained by applying a probabilistic mapping (embedding simulator) to c, obtaining thus the stego image $S_k(c)$ with S_0 being the identity map, $S_0(c) = c$ for all covers c. Technically, the embedding simulator depends on the key used for the simulator and the size of secret payload or embedding rate R, which will be measured in bits per pixel (bpp). This dependence is not made explicit to avoid cluttered notation.

We consider three different versions for our detector of diversified stego source:

1. Binary classifier (cover vs. all) trained on the cover class and the class of stego images embedded with all stego methods.
2. Multi-class detector classifying to $K + 1$ classes.
3. Bucket detector. First, K binary classifiers are trained to distinguish between covers and a specific stego method. Then, the front part of the networks before the IP layer is used as a feature extractor outputting a 512-dimensional feature to each input image. The features of all K detectors are then concatenated into one $512 \times K$ dimensional vector on which a multi-layered perceptron (MLP) is trained as either of the two detectors above.

For cover vs. all and the multi-class detector, the minibatches are formed by selecting cover-stego pairs $[c, S_k(c)]$, where both the covers c and $k \in \{1, \ldots, K\}$ are selected uniformly randomly.

The first approach is an embodiment of the "One Against All" approach to multi-classification. The logic is that a binary classifier presented with the class of cover images and images embedded with sufficiently many different stego schemes will be able to properly generalize to unseen stego methods with performance against known algorithms comparable to that of detectors dedicated to a single stego method. While this detector cannot identify a specific embedding scheme, it is easy to adjust its decision threshold to control the false alarm.

In contrast, the multi-class detector has the ability to identify the particular stego method. It can also be used for binary classification to detect between cover images and images with general steganographic content.

The bucket detector uses a concatenation of features extracted by networks trained for each embedding algo-

[1] Steganalysis of images of an arbitrary size with CNNs is studied in [8].

rithm. The logic is that perhaps a single network is not large enough to contain the complexity of a heavily diverse stego source, which may be alleviated by concatenating the features from each dedicated detector. The bucket detector can be trained as cover vs. all or as a multi-class. Since it internally uses a "feature representation," it could also be used for building a one-class detector, e.g., using one of the methods studied in [29].

A more elaborate version of the bucket detector can be obtained by training $\binom{K+1}{2}$ binary detectors to distinguish between every pair of classes. In this case, the concatenated "bucket" feature would have dimensionality $512\times\binom{K+1}{2}$. This version of the detector would be rather expensive to implement with deep learning based detectors unless K is relatively small.

For multi-class CNNs, the output layer of the SRNet is a soft-max applied to $K+1$ neurons that output the probability $q_k(x)$ of the input image x belonging to class k. The loss function evaluated for a minibatch of images $\mathcal{B}$ is

$$\begin{aligned} L(\mathcal{B}) &= -\frac{1}{|\mathcal{B}|}\sum_{x\in\mathcal{B}}\sum_{k=0}^{K} p_k(x)\log q_k(x) &(1)\\ &= \frac{1}{|\mathcal{B}|}\sum_{x\in\mathcal{B}} H(p(x),q(x)), &(2)\end{aligned}$$

where for image x, $p_k(x)$ is the ground truth pmf or an indicator function on $\{0,1,\ldots,K\}$ defined as $p_k(x)=1$ if x belongs to class k and zero otherwise. Since the cross-entropy $H(p,q)=H(p)+D_{\mathrm{KL}}(p||q)$, minimizing (2) is equivalent to minimizing the KL-divergence between a priori class probabilities and the class probabilities outputted by the network.

Denoting $\mathcal{B}_k=\{x\in\mathcal{B}|x=S_k(c)\text{ for some cover }c\}$, $k=0,1,\ldots,K$, the loss (2) can be written as

$$\begin{aligned} L(\mathcal{B}) &= \sum_{k=0}^{K}\frac{|\mathcal{B}_k|}{|\mathcal{B}|}\frac{1}{|\mathcal{B}_k|}\sum_{x\in\mathcal{B}_k} H(p(x),q(x)) \\ &\doteq \sum_{k=0}^{K}\pi_k\overline{H}_k &(3)\end{aligned}$$

for a sufficiently large minibatch, where $\overline{H}_k$ is the average cross-entropy for images from class k and $\pi_k=|\mathcal{B}_k|/|\mathcal{B}|$ is the prior probability that an image in batch $\mathcal{B}$ belongs to class k.

Training detectors for steganalysis requires pairing up each cover with the corresponding stego image because this pair constraint helps find the gradients separating the classes. Depending on how the pairs are formed, different prior π_k on each class is imposed. For example, pairing up two classes selected uniformly randomly from all $K+1$ classes (K stego methods and the cover class) leads to a loss function (3) with equal priors of each class, $\pi_k=1/(K+1)$ for all $k=0,1,\ldots,K$. On the other hand, forming pairs by first selecting a cover and then a stego method uniformly randomly leads to class priors:

$$\pi_0=\frac{1}{2},\ \pi_k=\frac{1}{2K},\ \text{for } k\geq 1, \tag{4}$$

This seems more appropriate for applications in steganalysis because the detector will have a lower false alarm. This way of forming the batches will be used in this paper.

We close this section by pointing out that the loss function could be adjusted by introducing weights into the indicator function $L_k(x)=w_kp_k(x)$ to incorporate other priors and/or further lower the false alarm or to improve the detection accuracy of a certain stego algorithm by forcing the optimizer to (asymptotically for large minibatches) minimize the following weighted loss:

$$L(\mathcal{B},\mathbf{w})=\sum_{k=0}^{K}w_k\pi_k\overline{H}_k. \tag{5}$$

The weights effectively translate to different priors $\pi'_k=w_k\pi_k/\sum_j w_j\pi_j$. For example, for $K=2$ stego schemes, using $w_0=L_0(c)=2$ for covers and $w_1=L_1(S_1(c))=w_2=L_2(S_2(c))=1$ for both stego classes changes the effective priors in the loss function from $\pi=\left(\frac{1}{2},\frac{1}{4},\frac{1}{4}\right)$ to $\pi'=\left(1,\frac{1}{4},\frac{1}{4}\right)/\left(1+\frac{1}{4}+\frac{1}{4}\right)=\left(\frac{2}{3},\frac{1}{6},\frac{1}{6}\right)$.

Training network detectors

All experiments in this paper were executed on images prepared from BOSSbase 1.01 [1] and BOWS2 [2] each with 10,000 grayscale images resized to 256×256 by 'imresize' in Matlab with default parameters. For training the CNN detectors, BOSSbase was randomly divided into three sets with 4,000 / 1,000 / 5,000 images. The 4,000 BOSSbase images were added to all 10,000 BOWS2 images to form the training set of 14,000 cover images ($2\times14,000$ images for training), $2\times1,000$ BOSSbase images for validation, and $2\times5,000$ BOSSbase images for testing. This dataset was adopted to be compatible with the datasets used in [40, 3].

All stego images were embedded with a fixed payload 0.4 bpp using embedding simulators operating on the corresponding rate–distortion bounds with the exception of the Edge Adaptive steganography [21]. This larger payload was selected to make the stego classes more separable since the stego source diversification increases the difficulty of detection. For non-adaptive LSB matching (LSBM), it was assumed that the message was embedded with an optimal ternary code, which translated to the change rate $\beta=H_3^{-1}(0.4)\doteq0.06254$, where $H_3(x)=-x\log_2 x-(1-x)\log_2(1-x)+x$ is the ternary entropy.

As explained in [3] and unless mentioned otherwise, the SRNet was trained with the stochastic gradient descend optimizer Adamax [18] with minibatches of 32 images (16 cover-stego pairs). The training database was shuffled after each epoch. Images in each batch were subjected to data augmentation with random mirroring and rotation of images by 90 degrees. The batch normalization parameters were learned via an exponential moving average with decay rate 0.9. The filter weights were initialized with the

He initializer[2] and 2×10^{-4} L2 regularization. Filter biases were set to 0.2 and no regularization. The weights in the fully connected classifier (IP) layer were initialized with a zero mean Gaussian with standard deviation 0.01 and no bias. All network detectors were trained for 400k iterations with learning rate (LR) 0.001 followed by 100k iterations with LR 0.0001. The snapshot achieving the best validation accuracy in the last 50k iterations was selected as the detector.

Detecting multiple stego algorithms

The results are divided into three subsections, each dedicated to one type of detector. The experiments with the bucket detector were scaled down because it performed poorly.

The cover vs. all binary detector and the multi-class detector were first trained for four content-adaptive steganographic algorithms WOW [12], S-UNIWARD [14], HILL [20], and MiPOD [33]. The networks were trained on three of these four to see how well they generalize to the fourth stego algorithm. At the same time, we tested the networks on LSBM to see how this non-adaptive and easy to detect algorithm is detected by a detector trained only on adaptive algorithms. Then, both detectors were trained on all four content-adaptive algorithms as well as all five, after adding LSBM. To assess the ability of the detector to generalize to unseen content-adaptive embedding, we tested them on Edge-Adaptive (EA) steganography [21] and on HUGO [24]. Finally, the detectors were trained on all seven embedding methods to see whether the network is capable to contain the complexity of this diverse stego source.

Cover vs. all stego detector

Table 1 shows the total probability of error P_{Err} of misdetecting each class when training on different combinations of three to five embedding algorithms listed in the first column. The errors of stego algorithms not included in training are highlighted in bold. The value P_{Err} for the cover class is the false-alarm probability, $P_{\mathrm{Err}} = P_{\mathrm{FA}}$, while P_{Err} corresponds to the missed detection, $P_{\mathrm{Err}} = P_{\mathrm{MD}}^{(k)}$, for stego class $k \in \{\mathrm{WOW}, \mathrm{SUNI}, \mathrm{HILL}, \mathrm{MiPOD}, \mathrm{LSBM}, \mathrm{EA}, \mathrm{HUGO}\}$, which is the probability of detecting the kth stego class as cover. The loss is the difference between the missed detection probability $P_{\mathrm{MD}}^{(k)}$ and the missed detection of a dedicated CNN trained on each individual stego method when adjusting its detection threshold to achieve the same false-alarm rate $P_{\mathrm{Err}} = P_{\mathrm{FA}}$ as the cover vs. all detector.

The largest loss was observed for HILL when not training on HILL. On the other hand, not training on MiPOD lead to an overall smallest loss across the four embedding schemes (HILL, MiPOD, WOW, and S-UNIWARD). The cover vs. all detector trained only on content-adaptive algorithms did poorly on the non-adaptive LSBM. Including LSBM in the training improved its detection dramatically even though the loss (0.0418) was the largest of the five stego methods. The detector trained on the four adaptive methods *and* LSBM, however, did not recognize stego images generated by EA and HUGO. The missed detection rate for EA was $P_{\mathrm{MD}}^{(\mathrm{EA})} = 0.2218$ (loss of 0.2142) and $P_{\mathrm{MD}}^{(\mathrm{HUGO})} = 0.5630$ for HUGO (loss of 0.4460). This indicates that the cover vs. all detector does not generalize well to previously unseen embedding methods.

As the next step, we added HUGO and EA to training to see how the accuracy of the cover vs. all detector scales w.r.t. the number of stego algorithms. The SRNet, however, did not converge from randomly initialized weights. This is most likely due to the small number of stego images from each class in a minibatch of 32 images. Since larger minibatches would not fit the memory of our Titan Xp GPUs (12 GB), we explored two different remedies: 1) seeding the network with weights from the cover vs. all detector trained for five stego algorithms and 2) training with a larger minibatch by employing the so-called gradient checkpointing[3] to trade off the memory for increased training time. The results are summarized in Table 2. By inspecting the loss, we conclude that training from scratch with the larger minibatch is better than seeding. Compared to the detector trained on five algorithms, the loss for the first five algorithms increased by 1.4–3.7%. The EA and HUGO experience the largest loss.

Multi-class detector

The performance of the multi-class detector was evaluated with confusion tables and with the probability of a miss P_{Err} interpreted for each stego class as the probability of not identifying the stego image as one of the stego algorithms, which is equal to the probability of identifying the stego image as cover.

Table 3 is an equivalent of Table 1 for the multi-class detector. By comparing both tables, we conclude that the multi-class detector provides better results than the binary cover vs. all detector but also fails to generalize to the non-adaptive LSBM and unseen adaptive algorithms (the miss probability for EA and HUGO were 0.3132 and 0.6744, respectively).

The confusion matrix when training the multi-class detector on five embedding algorithms (the columns of the table) is shown in Table 4. As before, the EA and HUGO algorithms are not detected well when excluded from training. However, when added to the set of known stego algorithms on which the detector is trained, all seven algorithms are reliably detected (Table 5) with losses ranging from 0.84% for MiPOD to 5.7% for EA (see Table 6). Please, note that the multi-class detector has a much smaller false alarm (0.0336) than the cover vs. all detector.

When training on all seven algorithms, the network would not converge, which we addressed by seeding it with the network trained on five algorithms and then by training from scratch with a larger minibatch of 64 images using gradient check-pointing as explained in the previous section. Both results are shown in Table 6, which again confirms that training from scratch with a larger minibatch is

[2] `https://arxiv.org/pdf/1502.01852v1.pdf`

[3] `https://github.com/openai/gradient-checkpointing`

	P_{Err}	Loss	P_{Err}	Loss	P_{Err}	Loss	P_{Err}	Loss	P_{Err}	Loss	P_{Err}	Loss
Cover	0.1374	0	0.1490	0	0.1562	0	0.1200	0	0.1392	0	0.1148	0
HILL	0.1742	0.0270	0.1516	0.0182	0.1578	0.0320	**0.2616**	**0.0960**	0.1762	0.0310	0.2190	0.0318
WOW	0.0688	0.0184	0.0890	0.0336	**0.0890**	**0.0452**	0.0840	0.0216	0.0872	0.0372	0.1080	0.0418
S-UNI	0.0884	0.0198	**0.1226**	**0.0620**	0.0934	0.0404	0.1028	0.0210	0.1042	0.0370	0.1108	0.0256
MiPOD	**0.2032**	**0.0376**	0.1520	0.0036	0.1532	0.0068	0.1928	0.0090	0.1716	0.0070	0.2070	0.0198
LSBM	**0.3754**	**0.3752**	**0.5466**	**0.5466**	**0.1672**	**0.1672**	**0.3394**	**0.3390**	**0.2612**	**0.2610**	0.0430	0.0418

Table 1. Missed detection probability P_{Err} on individual steganographic algorithms with the "cover vs. all" binary detector trained on multiple embedding algorithms. The loss is the increase of the error w.r.t. a detector dedicated to one stego source (algorithm) when adjusting its threshold to produce the same false alarm as the cover vs. all detector. Boldface marks the algorithms excluded during training.

	P_{Err}	Loss	P_{Err}	Loss
Cover	0.1106	0	0.1402	0
HILL	0.2704	0.0928	0.2132	0.0692
WOW	0.1392	0.0692	0.1164	0.0670
S-UNI	0.1392	0.0502	0.1136	0.0478
MiPOD	0.2508	0.0606	0.2046	0.0410
LSBM	0.0528	0.0516	0.0380	0.0378
EA	0.0932	0.0848	0.0810	0.0780
HUGO	0.2228	0.1022	0.1754	0.0828

Table 2. Missed detection probability P_{Err} on individual steganographic algorithms with the "cover vs. all" binary detector trained on all seven algorithms. Left: seeded with the detector trained on five algorithms, Right: trained from scratch with minibatch size 64 using gradient checkpointing. For a fair comparison, the loss is always w.r.t. a dedicated detector built to detect a specific embedding method at the same false-alarm rate as the cover vs. all detector.

preferable to seeding.

Bucket detector

Since the bucket detector performed the worst of the three studied approaches, we only comment on a few selected cases. Table 7 shows the missed detection rate when training the binary cover vs. all bucket detector on four and five algorithms, respectively. The cover vs. all detector trained as a single network (columns 10–13 in Table 1) provides a clear advantage over the bucket approach. In particular, note that including the LSBM in training did not sufficiently decrease the missed detection for LSBM.

Table 8 shows the confusion matrix when training the multi-class bucket detector on four embedding algorithms. This should be compared with the confusion matrix in Table 5 for the multi-class network detector trained on all seven algorithms. With a comparable false alarm rate, the detection of stego classes is markedly worse for the bucket detector despite being built for a less diverse stego source.

Comparison to prior art

To contrast the performance of the CNN-based detectors with prior art, we selected the low-complexity linear classifier [6] as the machine learning tool and the Spatial Rich Model (SRM) [7] for modeling images. In particular, we implemented a cover vs. all detector, a max-wins multi-class detector, and a MAP multi-class detector by modeling the projections of image features on weight vectors as a multi-variate Gaussian (MVG) distribution. These detectors were implemented to detect all seven embedding algorithms studied above at the same payload of 0.4 bpp. The training set was the union of the training and validation sets used for training the networks.

Cover vs. all detector

The cover vs. all detector was prepared by training one binary classifier on $N_{trn} = 15,000$ cover features and the same number of stego images with $[N_{trn}/K]$ images embedded by each of the K embedding algorithms. The decision threshold was set to obtain the same value of the false alarm as the CNN-based cover vs. all detector: 0.1106 (see Table 2).

Max-wins detector

For the max-wins multi-class detector, a binary classifier was trained between each pair of classes, giving us $n_w = \binom{K+1}{2} = 28$ classifiers. An image from the testing set is presented to all n_w classifiers, then the histogram of their answers is formed, and the bin with the maximum number of votes is the final detected class. Ties are resolved randomly. The decision thresholds for detectors trained for two stego classes were set to minimize the total probability of error under equal priors, $P_{\mathrm{E}} = \min_{P_{\mathrm{FA}}} \frac{1}{2}(P_{\mathrm{FA}} + P_{\mathrm{MD}})$. For the seven binary detectors between the cover class and a stego class, the thresholds were set to achieve the same false-alarm rate $P_{\mathrm{FA}} = \alpha_0$ across all seven detectors. A binary search for the value of α_0 was executed to obtain the same false-alarm rate of the max-wins detector as the CNN-based multi-class detector: 0.0336 (see Table 5).

Multi-class MAP detector

The third type of detector was inspired by [5] where the authors modeled the projections of a rich feature on weight vectors of base learners in the FLD-ensemble [19] with a multi-variate Gaussian distribution. The shift hypothesis stating that embedding changes the mean of the MVG of covers but not its covariance allowed considering a novel criteria of optimality – the minimax criterion that maximizes the worst correct stego class probability for a prescribed false alarm rate – probability of detecting a cover image as one of the stego classes (see Theorem 1 in [5]). Since the output of detectors built as CNNs is non-Gaussian (see, e.g., the ROC curves in [3]), the detectors proposed in this paper cannot be built in the same fashion. Instead, to compare both detectors, we used the modeling

	P_{Err}	Loss	P_{Err}	Loss	P_{Err}	Loss	P_{Err}	Loss	P_{Err}	Loss	P_{Err}	Loss
Cover	0.0578	0	0.0682	0	0.0684	0	0.0566	0	0.0578	0	0.0590	0
HILL	0.2490	−0.0016	0.2348	−0.0048	0.2364	0.0066	**0.3314**	**0.0788**	0.2534	0.0028	0.2634	0.0162
WOW	0.1338	0.0112	0.1368	0.0290	**0.1504**	**0.0426**	0.1330	0.0098	0.1330	0.0104	0.1516	0.0310
S-UNI	0.1606	0.0098	**0.2226**	**0.0862**	0.1592	0.0226	0.1576	0.0052	0.1508	0.0000	0.1724	0.0238
MiPOD	**0.3180**	**0.0584**	0.2544	0.0134	0.2496	0.0080	0.2530	−0.0092	0.2508	−0.0088	0.2710	0.0130
LSBM	**0.4244**	**0.4172**	**0.7166**	**0.7120**	**0.3486**	**0.3440**	**0.2758**	**0.2678**	**0.4266**	**0.4186**	0.0356	0.0288

Table 3. Missed detection probability P_{Err} on individual steganographic algorithms with the multi-class detector trained on different combinations of embedding algorithms. The loss is the increase of the error w.r.t. a detector dedicated to one stego source (algorithm). Boldface marks the algorithms excluded during training and listed in the top row. When tested on individual algorithms, the output of the multi-class detector was considered correct when the stego image was detected as one of the stego methods.

True\Det	Cover	HILL	WOW	S-UNI	MiPOD	LSBM
Cover	0.9410	0.0196	0.0062	0.0102	0.0160	0.0070
HILL	0.2634	0.6472	0.0142	0.0226	0.0472	0.0054
WOW	0.1516	0.0250	0.7146	0.0786	0.0256	0.0046
S-UNI	0.1724	0.0366	0.0578	0.6764	0.0440	0.0128
MiPOD	0.2710	0.0840	0.0260	0.0464	0.5654	0.0072
LSBM	0.0356	0.0032	0.0034	0.0068	0.0004	0.9506
EA	0.3132	0.0434	0.0850	0.2008	0.2292	0.1284
HUGO	0.6744	0.0838	0.0158	0.1026	0.1070	0.0164

Table 4. Confusion matrix for the multi-class detector trained on five embedding algorithms. The last two rows show the results on EA and HUGO stego images that were not used for training.

True\Det	Cover	HILL	WOW	S-UNI	MiPOD	LSBM	EA	HUGO
Cover	0.9664	0.0078	0.0034	0.0024	0.0084	0.0010	0.0030	0.0066
HILL	0.3304	0.5962	0.0116	0.0102	0.0346	0.0018	0.0056	0.0096
WOW	0.1962	0.0192	0.7004	0.0514	0.0210	0.0012	0.0072	0.0034
S-UNI	0.2182	0.0306	0.0656	0.6150	0.0396	0.0082	0.0072	0.0156
MiPOD	0.3332	0.0766	0.0194	0.0326	0.5190	0.0026	0.0068	0.0098
LSBM	0.0596	0.0010	0.0010	0.0088	0.0008	0.9248	0.0020	0.0020
EA	0.1134	0.0030	0.0026	0.0020	0.0020	0.0002	0.8726	0.0042
HUGO	0.3102	0.0142	0.0048	0.0116	0.0094	0.0020	0.0162	0.6316

Table 5. Confusion matrix for the multi-class detector when training on all seven embedding algorithms.

assumption imposed on projections of rich features to build a detector that mimics the objective function minimized in the corresponding CNN detectors – the total detection error with class priors (4). We now elaborate on the details of this approach.

We will assume that images are represented with feature vectors $\mathbf{f} \in \mathbb{R}^d$, where d is the feature dimensionality (for the SRM, $d = 34,671$). We split the index set of the training set into two disjoint subsets $\mathcal{I}_1$ and $\mathcal{I}_2$ with 12,000 and 3,000 cover images and the same number of stego images from each class. The first is used to train all n_w binary classifiers between classes k and l with weight vectors $\mathbf{w}^{(k,l)} \in \mathbb{R}^d$. We remind that for $K = 7$ stego classes there will be $n_w = 28$ of such weight vectors. The images from the second subset will be used to estimate the means $\boldsymbol{\mu}^{(k)} \in \mathbb{R}^{n_w}$, $k = 0, 1, \ldots, K$, and the covariance matrices $\mathbf{C}^{(k)} \in \mathbb{R}^{n_w \times n_w}$ of the MVG distributions $\mathcal{N}(\boldsymbol{\mu}^{(k)}, \mathbf{C}^{(k)})$ of the n_w-dimensional vector of projections of feature vectors from class k on all n_w weight vectors. In other words, there are $K+1$ MVG distributions in the n_w-dimensional space of projections. Putting all n_w weight vectors as rows of a matrix $\mathbf{W} \in \mathbb{R}^{n_w \times d}$ and denoting the sets of features $\mathbf{f}$ from class k and training set $\mathcal{I}_2$ as $\mathcal{F}^{(k)}_{\mathcal{I}_2}$, the sample means

	P_{Err}	Loss	P_{Err}	Loss
Cover	0.0336	0	0.0644	0
HILL	0.3304	0.0312	0.2544	0.0190
WOW	0.1962	0.0392	0.1402	0.0270
S-UNI	0.2182	0.0344	0.1560	0.0158
MiPOD	0.3332	0.0236	0.2564	0.0084
LSBM	0.0596	0.0398	0.0476	0.0428
EA	0.1134	0.0610	0.0834	0.0572
HUGO	0.3102	0.0946	0.2020	0.0280

Table 6. Missed detection probability P_{Err} and the loss w.r.t. dedicated detectors for the multi-class detector trained on all seven embedding algorithms. Left: seeded with the detector trained on five algorithms, Right: trained from scratch with mini-batch size 64 using gradient checkpointing. For a fair comparison, the loss is always w.r.t. a dedicated detector built to detect a specific embedding method at the same false-alarm rate as the cover vs. all detector.

	P_{Err}	Loss
Cover	0.2537	0
HILL	0.1793	0.0749
WOW	0.0953	0.0844
S-UNI	0.1460	0.0964
MiPOD	0.1737	0.0650
LSBM	0.5983	0.3954

	P_{Err}	Loss
Cover	0.2050	0
HILL	0.2453	0.0829
WOW	0.1430	0.0874
S-UNI	0.1857	0.0919
MiPOD	0.2550	0.0750
LSBM	0.2510	0.2041

Table 7. Missed detection probability P_{Err} and the loss w.r.t. dedicated detectors for the binary bucket detector trained on four (left) and five (right) algorithms.

and covariances are obtained through

$$\boldsymbol{\mu}^{(k)} = \mathbb{E}\left[\mathbf{W} \cdot \mathbf{f} \middle| \mathbf{f} \in \mathcal{F}^{(k)}_{\mathcal{I}_2}\right], \tag{6}$$

$$\mathbf{C}^{(k)}_{kl} = \mathbb{E}\left[\left((\mathbf{W} \cdot \mathbf{f})_k - \boldsymbol{\mu}^{(k)}_k\right) \cdot \left((\mathbf{W} \cdot \mathbf{f})_l - \boldsymbol{\mu}^{(k)}_l\right) \middle| \mathbf{f} \in \mathcal{F}^{(k)}_{\mathcal{I}_2}\right]. \tag{7}$$

Given a feature $\mathbf{f} \in \mathbb{R}^d$ of a test image, the hypothesis test is

$$\begin{aligned}
\mathrm{H}_0 &: \mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}^{(0)}, \mathbf{C}^{(0)}) \\
\mathrm{H}_1 &: \mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}^{(1)}, \mathbf{C}^{(1)}) \\
&\ldots\ldots \\
\mathrm{H}_K &: \mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}^{(K)}, \mathbf{C}^{(K)}).
\end{aligned}$$

True\Detected	Cover	HILL	WOW	S-UNI	MiPOD
Cover	0.9660	0.0070	0.0077	0.0107	0.0087
HILL	0.4653	0.3980	0.0353	0.0353	0.0660
WOW	0.3690	0.0666	0.3893	0.0913	0.0837
S-UNI	0.4650	0.0530	0.1143	0.2643	0.1033
MiPOD	0.5020	0.0977	0.0783	0.0797	0.2423
LSBM	0.8527	0.0220	0.0307	0.0660	0.2867

Table 8. Confusion matrix for the bucket multi-class detector trained on four embedding schemes.

With prior class probabilities $\pi_l = \Pr(\mathrm{H}_l)$, $l = 0, 1, \ldots, K$, the MAP multi-class detector will decide class k when

$$k = \arg\max_{k'} \Pr(\mathrm{H}_{k'}|\mathbf{f}) = \arg\max_{k'} \Pr(\mathbf{f}|\mathrm{H}_{k'})\Pr(\mathrm{H}_{k'}), \tag{8}$$

where

$$\Pr(\mathbf{f}|\mathrm{H}_l) = \left((2\pi)^{n_w}|\mathbf{C}^{(l)}|\right)^{-1/2} \times \exp\left(\frac{1}{2}(\mathbf{W}\cdot\mathbf{f} - \boldsymbol{\mu}^{(l)})^T(\mathbf{C}^{(l)})^{-1}(\mathbf{W}\cdot\mathbf{f} - \boldsymbol{\mu}^{(l)})\right) \tag{9}$$

are the estimated MVG densities.

To allow direct comparison, the parameter π_0 in the class priors $\left(\pi_0, \frac{1-\pi_0}{K}, \ldots, \frac{1-\pi_0}{K}\right)$ was selected so that the MAP detector achieves the same false-alarm rate as the CNN-based multi-class detector: 0.0336 (see Table 5).

The estimated covariance matrices for all eight classes are shown (after scaling them to the same dynamic range) in Figure 1. While the shift hypothesis was observed to approximately hold for modern content-adaptive algorithms, the covariance of LSBM exhibited a completely different structure. Also, despite "structural similarity" of the covariance matrix of EA to Covers and modern adaptive schemes, its entries were by an order of magnitude larger than those of the five remaining content-adaptive schemes and Covers. This is most likely due to the rather large average change rate (≈ 0.16) of EA compared to ≈ 0.09 for modern adaptive embedding. This is why in the MAP detector, we did *not* use the shift hypothesis and estimated the covariance matrices for each class separately. This finding also precludes constructing detectors with the minimax criterion as proposed in [5]. To obtain further insight into the distribution of feature projections, in Figure 2 we show the means of the MVG distributions when scaling the $n_w = 28$ dimensional space of projections to three dimensions using multi-dimensional scaling in Matlab. As expected, HILL, MiPOD, and S-UNIWARD classes are closest to Covers with WOW and HUGO at a larger distance, with the two most detectable classes, EA and LSBM, lying the furthest from Covers.

Results

All three detectors in this section were trained on the same split of the dataset into the training and testing set as for the network detectors. The cover vs. all was trained on all $2 \times 15,000$ images from the training set. The max-wins and the MAP detectors were trained on 12,000 randomly selected cover images and the corresponding stego images. The remaining 3,000 cover (and stego) images were used to estimate the decision thresholds for the individual n_w classifiers and the MVG parameters.

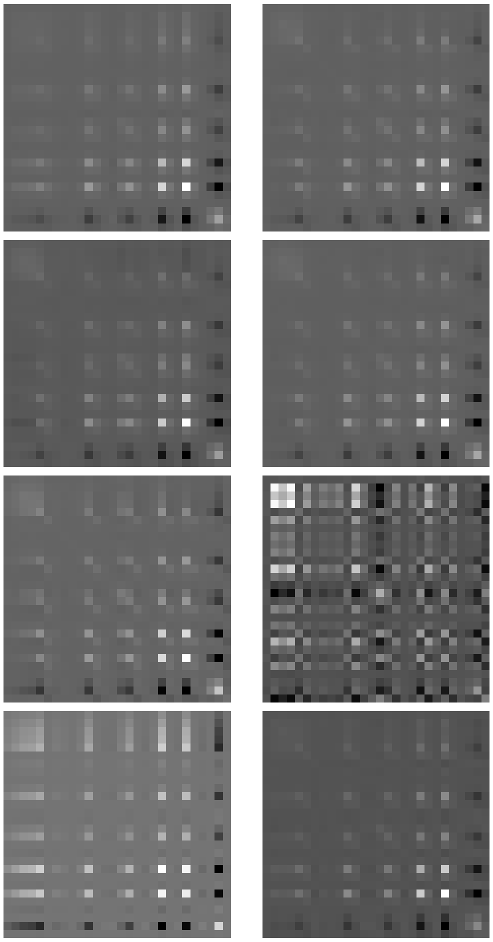

Figure 1. Covariance matrices for projections of feature vectors from (by rows): cover, HILL, WOW, S-UNIWARD, MiPOD, LSBM, EA, and HUGO. The matrices were scaled to 8-bit grayscale images for visualization.

Table 9 contrasts the the probability of a miss, P_{Err}, for the cover vs. all detector built with a CNN (Table 2) and with a linear classifier on SRM features. While the overall detectability of stego algorithms seems to exhibit a similar pattern, the CNN detector is markedly more accurate.

For better compactness, instead of presenting the complete confusion tables, Table 10 shows the probability of a miss P_{Err} and the correct classification probability P_{CC} (the diagonal of the confusion matrix) for three multi-class detectors: the CNN from the previous section, the max-wins, and MAP detectors implemented with SRM features. While the MAP detector is clearly better than the max-wins (except for EA), they are outperformed by the CNN both in terms of a significantly lower P_{Err} and larger P_{CC}.

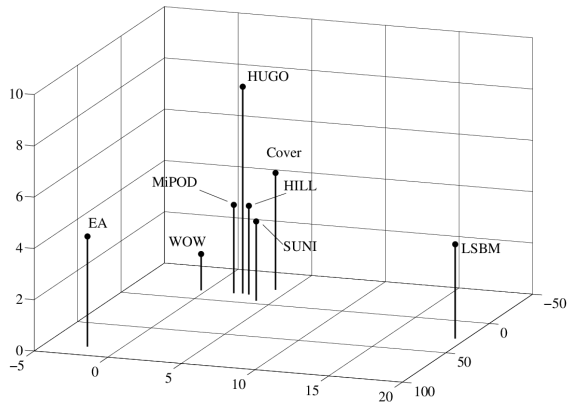

Figure 2. *Visualizing the means of the MVG distributions in the space of projections when mapped from $n_w = 28$ dimensions to 3 using multi-dimensional scaling.*

	CNN		SRM	
	P_{Err}	Loss	P_{Err}	Loss
Cover	0.1106	0	0.1106	0
HILL	0.2704	0.0928	0.5940	0.4164
WOW	0.1392	0.0692	0.4254	0.3554
S-UNI	0.1392	0.0502	0.4112	0.3222
MiPOD	0.2508	0.0606	0.5396	0.3494
LSBM	0.0528	0.0516	0.1946	0.1934
EA	0.0932	0.0848	0.2256	0.2172
HUGO	0.2228	0.1022	0.3860	0.2654

Table 9. Missed detection probability P_{Err} and loss on individual steganographic algorithms achieved with the "cover vs. all" binary detector implemented as a CNN (left) and a low-complexity linear classifier with SRM features (right) on all seven algorithms (the CNN detector was taken from Table 2).

Conclusions

This paper deals with the problem of detecting secrets potentially embedded with many different content-adaptive and non-adaptive steganographic algorithms. The best detector was a multi-class convolutional neural network implemented with the previously proposed deep residual architecture called SRNet. We showed how its loss function can be adjusted to control the false alarms or missed detection of selected stego schemes. When trained on seven embedding algorithms, this multi-class detector was able to reliably classify the stego algorithm, while its ability to detect steganographic content decreased only marginally w.r.t. binary CNN detectors dedicated (and tested) on a specific embedding algorithm. Also investigated were detectors built as cover vs. all and bucket detectors in the "feature space" outputted by dedicated binary classifiers trained for each embedding scheme. The multi-class detector also performed significantly better than detectors constructed as linear classifiers while representing images with the SRM.

	Multi-class CNN		Max-wins (SRM)		MAP (SRM)	
	P_{Err}	P_{CC}	P_{Err}	P_{CC}	P_{Err}	P_{CC}
Cover	0.0336	0.9664	0.0332	0.9668	0.0336	0.9664
HILL	0.3304	0.5962	0.8232	0.0570	0.7044	0.1702
WOW	0.1962	0.7004	0.6540	0.2248	0.6098	0.3014
S-UNI	0.2182	0.6150	0.6684	0.1788	0.5550	0.2768
MiPOD	0.3332	0.5190	0.7676	0.0792	0.6850	0.1332
LSBM	0.0596	0.9248	0.2896	0.6756	0.2322	0.6386
EA	0.1134	0.8726	0.2606	0.6958	0.3934	0.5680
HUGO	0.3102	0.6316	0.5256	0.3984	0.4684	0.4736

Table 10. Probability of miss P_{Err} and the correct classification probability P_{CC} for the multi-class CNN based detector (left), the max-wins detector implemented with SRM (middle), and the MAP detector with MVG model of weight vector projections (right).

While the multi-class CNN was able to "contain" the complexity of the diversified stego source in the sense that it provided detection of steganography comparable to that of dedicated detectors, it appeared to struggle to recognize previously unseen steganographic methods. The difficult problem of building a universal blind steganalyzer is thus postponed to our future work.

This study is limited to a rather narrow source – the union of the popular BOSSbase and BOWS2 – and to stego images embedded with a fixed payload. A mismatch in the cover source is likely to significantly decrease the detection performance. Battling the cover source mismatch in spatial-domain steganography is an extraordinarily difficult problem due to the great diversity of possible processing that can be applied to images prior to embedding. Based on previous studies, the impact of the mismatch is likely to be significantly smaller in the JPEG domain where we expect the multi-class detector to be more universal. At the moment, the detectors constructed in this paper can likely be trusted only when the analyst has access to the cover source to generate enough training examples. Detection of diversified stego sources in the JPEG domain will be the subject of our future research.

Finally, we note that although we fixed the relative payload, we expect the general satisfactory performance to transfer to an unknown payload. Based on our preliminary experiments, training a dedicated CNN-based detector on a "mid payload" produced comparable accuracy as training on the correct random mixture of payloads. This problem, too, is expected to make its way into our future effort.

All code used to produce the results in this paper, including the network configuration files will be available from `http://dde.binghamton.edu/download/`.

Acknowledgments

The work on this paper was supported by NSF grant No. 1561446 and by DARPA under agreement number FA8750-16-2-0173. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of DARPA or the U.S. Government.

References

[1] P. Bas, T. Filler, and T. Pevný. Break our steganographic system – the ins and outs of organizing BOSS. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, volume 6958 of Lecture Notes in Computer Science, pages 59–70, Prague, Czech Republic, May 18–20, 2011.

[2] P. Bas and T. Furon. BOWS-2. `http://bows2.ec-lille.fr`, July 2007.

[3] M. Boroumand, M. Chen, and J. Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 2018. Under review.

[4] M. Chen, V. Sedighi, M. Boroumand, and J. Fridrich. JPEG-phase-aware convolutional neural network for steganalysis of JPEG images. In M. Stamm, M. Kirchner, and S. Voloshynovskiy, editors, *The 5th ACM Workshop on Information Hiding and Multimedia Security*, Philadelphia, PA, June 20–22, 2017.

[5] R. Cogranne and J. Fridrich. Modeling and extending the ensemble classifier for steganalysis of digital images using hypothesis testing theory. *IEEE Transactions on Information Forensics and Security*, 10(2):2627–2642, December 2015.

[6] R. Cogranne, V. Sedighi, T. Pevný, and J. Fridrich. Is ensemble classifier needed for steganalysis in high-dimensional feature spaces? In *IEEE International Workshop on Information Forensics and Security*, Rome, Italy, November 16–19, 2015.

[7] J. Fridrich and J. Kodovský. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, June 2011.

[8] C. Fuji-Tsang and J. Fridrich. Steganalyzing images of arbitrary size with CNNs. San Francisco, CA, 2018.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 27–30 2016.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision ECCV*, volume 9908 of Lecture Notes in Computer Science, Amsterdam, Octoer 8–16 2016.

[11] S. Hetzl and P. Mutzel. A graph-theoretic approach to steganography. In J. Dittmann, S. Katzenbeisser, and A. Uhl, editors, *Communications and Multimedia Security, 9th IFIP TC-6 TC-11 International Conference, CMS 2005*, volume 3677 of Lecture Notes in Computer Science, pages 119–128, Salzburg, Austria, September 19–21, 2005.

[12] V. Holub and J. Fridrich. Designing steganographic distortion using directional filters. In *Fourth IEEE International Workshop on Information Forensics and Security*, Tenerife, Spain, December 2–5, 2012.

[13] V. Holub and J. Fridrich. Phase-aware projection model for steganalysis of JPEG images. In A. Alattar and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2015*, volume 9409, San Francisco, CA, February 8–12, 2015.

[14] V. Holub, J. Fridrich, and T. Denemark. Universal distortion design for steganography in an arbitrary domain. *EURASIP Journal on Information Security, Special Issue on Revised Selected Papers of the 1st ACM IH and MMS Workshop*, 2014:1, 2014.

[15] G. Huang, Z. Liu, L. van der Maaten, and K. Q Weinberger. Densely connected convolutional networks. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, July 21–26, 2017.

[16] A. D. Ker. Batch steganography and pooled steganalysis. In J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, editors, *Information Hiding, 8th International Workshop*, volume 4437 of Lecture Notes in Computer Science, pages 265–281, Alexandria, VA, July 10–12, 2006. Springer-Verlag, New York.

[17] A. D. Ker and T. Pevný. The steganographer is the outlier: Realistic large-scale steganalysis. *IEEE Transactions on Information Forensics and Security*, 9(9):1424–1435, September 2014.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, 2014. http://arxiv.org/abs/1412.6980.

[19] J. Kodovský, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security*, 7(2):432–444, April 2012.

[20] B. Li, M. Wang, and J. Huang. A new cost function for spatial image steganography. In *Proceedings IEEE, International Conference on Image Processing, ICIP*, Paris, France, October 27–30, 2014.

[21] W. Luo, F. Huang, and J. Huang. Edge adaptive image steganography based on LSB matching revisited. *IEEE Transactions on Information Forensics and Security*, 5(2):201–214, June 2010.

[22] A. Munoz and J. M. Moguerza. Estimation of high-density regions using one-class neighbor machines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):476–480, 2006.

[23] T. Pevný. Detecting messages of unknown length. In A. Alattar, N. D. Memon, E. J. Delp, and J. Dittmann, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security and Forensics III*, volume 7880, pages OT 1–12, San Francisco, CA, January 23–26, 2011.

[24] T. Pevný, T. Filler, and P. Bas. Using high-dimensional image models to perform highly undetectable steganography. In R. Böhme and R. Safavi-Naini, editors, *Information Hiding, 12th International Conference*, volume 6387 of Lecture Notes in Computer Science, pages 161–177, Calgary, Canada, June 28–30, 2010. Springer-Verlag, New York.

[25] T. Pevný and J. Fridrich. Towards multi-class blind steganalyzer for JPEG images. In M. Barni, I. J. Cox, T. Kalker, and H. J. Kim, editors, *International Workshop on Digital Watermarking*, volume 3710 of Lecture Notes in Computer Science, Siena, Italy, September

15–17, 2005. Springer-Verlag, Berlin.

[26] T. Pevny and J. Fridrich. Determining the stego algorithm for JPEG images. *Special Issue of IEE Proceedings on Information Security*, 153(3):75–139, 2006.

[27] T. Pevný and J. Fridrich. Multiclass blind steganalysis for JPEG images. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, pages O 1–O 13, San Jose, CA, January 16–19, 2006.

[28] T. Pevný and J. Fridrich. Merging Markov and DCT features for multi-class JPEG steganalysis. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 3 1–14, San Jose, CA, January 29–February 1, 2007.

[29] T. Pevný and J. Fridrich. Novelty detection in blind steganalysis. In A. D. Ker, J. Dittmann, and J. Fridrich, editors, *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 167–176, Oxford, UK, September 22–23, 2008.

[30] N. Provos. Defending against statistical steganalysis. In *10th USENIX Security Symposium*, pages 323–335, Washington, DC, August 13–17, 2001.

[31] Y. Qian, J. Dong, W. Wang, and T. Tan. Deep learning for steganalysis via convolutional neural networks. In A. Alattar and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2015*, volume 9409, San Francisco, CA, February 8–12, 2015.

[32] P. Sallee. Model-based methods for steganography and steganalysis. *International Journal of Image Graphics*, 5(1):167–190, 2005.

[33] V. Sedighi, R. Cogranne, and J. Fridrich. Content-adaptive steganography by minimizing statistical detectability. *IEEE Transactions on Information Forensics and Security*, 11(2):221–234, 2016.

[34] I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2001. Available electronically at `http://www.jmlr.org/papers/volume2/steinwart01a/steinwart01a.ps.gz`.

[35] D. Upham. Steganographic algorithm JSteg. Software available at http://zooid.org/ paul/crypto/jsteg.

[36] A. Westfeld. High capacity despite better steganalysis (F5 – a steganographic algorithm). In I. S. Moskowitz, editor, *Information Hiding, 4th International Workshop*, volume 2137 of Lecture Notes in Computer Science, pages 289–302, Pittsburgh, PA, April 25–27, 2001. Springer-Verlag, New York.

[37] G. Xu. Deep convolutional neural network to detect J-UNIWARD. In M. Stamm, M. Kirchner, and S. Voloshynovskiy, editors, *The 5th ACM Workshop on Information Hiding and Multimedia Security*, Philadelphia, PA, June 20–22, 2017.

[38] G. Xu, H. Z. Wu, and Y. Q. Shi. Structural design of convolutional neural networks for steganalysis. *IEEE Signal Processing Letters*, 23(5):708–712, May 2016.

[39] J. Yang, Y.-Q. Shi, E.K. Wong, and X. Kang. JPEG steganalysis based on densenet. *CoRR*, abs/1711.09335, 2017.

[40] J. Ye, J. Ni, and Y. Yi. Deep learning hierarchical representations for image steganalysis. *IEEE Transactions on Information Forensics and Security*, 12(11):2545–2557, November 2017.

[41] J. Zeng, S. Tan, B. Li, and J. Huang. Large-scale JPEG image steganalysis using hybrid deep-learning framework. *IEEE Transactions on Information Forensics and Security*, 13(5):1200–1214, 2018.

Author Biography

Jan Butora is currently a PhD student in Electrical and Computer Engineering at Binghamton University. His research interest are in the field of media security and forensics, and, particularly steganography and steganalysis of digital images.

Jessica Fridrich is Distinguished Professor of Electrical and Computer Engineering at Binghamton University. She received her PhD in Systems Science from Binghamton University in 1995 and MS in Applied Mathematics from Czech Technical University in Prague in 1987. Her main interests are in steganography, steganalysis, and digital image forensics. Since 1995, she has received 20 research grants totaling over $12 mil that lead to more than 200 papers and 7 US patents.