Reference Channels for Steganalysis of Images with Convolutional Neural Networks

Mo Chen, Mehdi Boroumand, and Jessica Fridrich
Department of Electrical and Computer Engineering, Binghamton University, NY, USA
Email: mochen8@gmail.com, {mboroum1,fridrich}@binghamton.edu

ABSTRACT

When available, reference signals may dramatically improve the accuracy of steganalysis. Particularly powerful reference signals are embedding invariants that exist when the steganographic algorithm swaps values from small disjoint subsets of the cover elements' dynamic range, such as, but not limited to, embedding schemes utilizing least significant bit replacement. This paper describes a general method how to prepare such reference signals for a certain type of embedding operations, and incorporate them in detectors built as convolutional networks to improve their detection accuracy. The beneficial effect of reference signals is shown experimentally in both the spatial and especially JPEG domain, on model-based steganography and a generic LSB flipper with and without stochastic restoration of the histogram (OutGuess).

KEYWORDS

Steganography, steganalysis, convolutional neural network, parity, reference

ACM Reference Format:
Mo Chen, Mehdi Boroumand, and Jessica Fridrich. 2019. Reference Channels for Steganalysis of Images with Convolutional Neural Networks. In *ACM Information Hiding and Multimedia Security Workshop (IHMMSec '19), July 3–5, 2019, TROYES, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3335203.3335733

1 INTRODUCTION

Steganography is the art of communicating covertly by hiding the very existence of the message in a host signal called the cover object. Modern steganographic methods for covers in the form of digital images embed the secret by slightly modifying individual pixel values or the quantized discrete cosine transform (DCT) coefficients in a JPEG file. While imperceptible to human senses, the modifications may introduce characteristic statistical anomalies revealing the presence of the secret. Identifying such anomalies is the subject of steganalysis, the art of building detectors of steganography.

IHMMSec '19, July 3–5, 2019, TROYES, France

ACM ISBN 978-1-4503-6821-6/19/06…$15.00
https://doi.org/10.1145/3335203.3335733

Steganalysis can generally be improved if the detector is supplied with additional reference information about the cover image. Examples are calibration signals in the form of a decompressed-cropped-recompressed image [16, 20], a downsampled image [13, 19], or an image of a different scene that partially overlaps with the analyzed image [30]. An especially powerful reference signal becomes available if the embedding changes can be "erased" in the sense that an operation exists that does not change the image much but, when applied to both cover and stego versions of the same image, it produces the same result. Such embedding invariants exist, for example, when the sender overwrites the least significant bits (LSBs) of cover elements with message bits – the image with all its LSBs set to zero is an example of an embedding invariant. This reference signal is fundamentally responsible for the existence of many powerful attacks on LSB replacement (LSBR), be it the so-called structural attacks [6, 9, 12, 14] or detectors derived with statistical signal detection tools [5, 7, 8, 15, 26, 27, 37]. Feature-based detectors implemented using machine learning can also use embedding invariants by augmenting the feature extracted from the image with the feature extracted from the embedding invariant [11].

The reference signal utilized in this paper exists when the embedding changes are constrained to swaps of small disjoint groups of cover values. Typical examples are schemes that replace one or more LSBs of cover elements with message bits or the operation employed in Model-Based Steganography (MBS) for JPEG images [23], which is not constrained to flipping LSBs. Although fundamentally insecure, LSB replacement is the most popular type of steganography because it is simple and can be applied to virtually any sampled signal. As of October 2017, out of 2863 tools available on the Internet capable of hiding data in digital images, 1024 (36%) of them embed secrets by manipulating LSBs.[1]

While detection of the early "naive" embedding paradigms has been studied in the past to a great extent, and while the concensus among researchers might be that the detection of said embedding is as good as it can ever be, as this paper shows, there is still much room for improvement, especially for small secret payloads and for embedding algorithms that either do not merely flip bits but allow reference signals, such as MBS. Improved detection of short secret messages should certainly be of interest to practitioners given the large number of publicly available software tools that manipulate LSBs.

Recently, deep learning with convolutional neural networks (CNNs) has led to great advances in steganalysis by jointly optimizing the feature design and the detector [2, 3, 18, 22, 32–36].

[1] N. Johnson, "IoT Forensic Considerations and Steganography Beyond Images." Invited talk presented in the Network and Cloud Forensics Workshop, IEEE Conference on Communications and Network Security, October 9–11, 2017, Las Vegas, Nevada, USA.

Because the first operation in a CNN is a convolution, such detectors do not explicitly "see" parity of pixels and thus cannot easily discover the existence of embedding invariants. In this paper, we show how to prepare the embedding invariant for a more general class of embedding operations, a generalization of LSB flipping, and how to incorporate this invariant in detectors built as CNNs. In JPEG domain, the construction of the reference signal requires some care because of the properties of first-order statistics of DCT coefficients and the type of the embedding operation.

In the next section, we describe the process of creating a useful reference signal for general embedding operations with a swapping constraint. A previously proposed deep residual network for steganalysis of digital images called SRNet [2] is modified in Section 3 to make use of the reference signal. The setup of our empirical study and the common core of all experiments appear in Section 4. In Section 5, we report the results of experiments with LSBR in the spatial domain when steganalyzing with both single-channel and two-channel SRNet and YeNet [34], and contrast their performance to previous art – rich models utilizing the same reference signal and the weighted stego-image detector. The filters learned in the first convolutional layer of both network detectors are analyzed to obtain some insight into how CNNs utilize the reference channel. Detection of JPEG domain steganographic algorithms with reference channels is investigated in Section 6. We take a look at two types of reference channels and analyze the filters learned in the first layer. The performance of the two-channel SRNet is contrasted with rich models (JRM [17] and GFR [25]) augmented with features extracted from the same reference signal as well as detectors built using statistical signal detection tools [26, 27]. The paper is concluded in Section 7.

2 REFERENCE SIGNALS

While our study is restricted to 8-bit grayscale images, the proposed methodology is applicable to a much wider range of cover objects and formats. A cover image will be represented with an $N_1 \times N_2$ matrix $\mathbf{x} = (x_{ij}) \in \mathcal{I}^{N_1 \times N_2}$ while reserving $\mathbf{y} = (y_{ij}) \in \mathcal{I}^{N_1 \times N_2}$ for the corresponding stego image, where $\mathcal{I}$ is the dynamic range. For 8-bit grayscale images in the spatial domain, $\mathcal{I} = \{0, \dots, 255\}$, while for a JPEG image, $\mathcal{I} = \{-1024, \dots, 1023\}$.

2.1 Swapping constraint

An embedding invariant (reference signal) is obtained using a mapping $\mathcal{R} : \mathcal{I}^{N_1 \times N_2} \to \mathcal{I}^{N_1 \times N_2}$ such that $\mathcal{R}(\mathbf{x}) = \mathcal{R}(\mathbf{y})$ for any cover image $\mathbf{x}$ and all its possible stego versions $\mathbf{y}$. For an embedding invariant to be useful for steganalysis, however, $\mathcal{R}(\mathbf{y})$ needs to be close to $\mathbf{x}$. For example, one can always obtain trivial (and useless) invariants with $\mathcal{R}(\mathbf{x}) = \mathbf{r} \in \mathcal{I}^{N_1 \times N_2}$, $\mathbf{r}$ chosen arbitrarily but fixed.

The type of the reference signal investigated in this paper generally exists when the steganographic algorithm restricts the embedding modifications to swapping values within M mutually disjoint subsets $\mathcal{I}_1, \dots, \mathcal{I}_M$ of $\mathcal{I}$, $\mathcal{I} = \mathcal{I}_1 \cup \cdots \cup \mathcal{I}_M$, $\mathcal{I}_k \cap \mathcal{I}_l = \emptyset$, $k \neq l$.[2] In this case, the embedding modifications satisfy the following swapping constraint

$$x_{ij} \in \mathcal{I}_k \iff y_{ij} \in \mathcal{I}_k, \text{ for all } i, j. \tag{1}$$

[2]Technically, the subsets may not necessarily be all of the same cardinality.

For example, LSBR overwrites the LSBs of selected pixels with message bits, which means that the dynamic range of pixels is split into disjoint pairs $\mathcal{I}_k = \{2k, 2k+1\}$, $k = 0, \dots, 127$. If the embedding overwrites two LSBs with message bits, e.g., for increased embedding capacity, the changes are restricted to disjoint quadruples of values $\mathcal{I}_k = \{4k, 4k+1, 4k+2, 4k+3\}$, $k = 0, \dots, 63$.

As another example, we point out the embedding operation of model-based steganography (MBS) [23], which restricts its modifications to pairs of values that do not differ only in their LSBs:

$$\mathcal{I}_k = \begin{cases} \{2k-1, 2k\} & k \geq 1 \\ \{2k, 2k+1\} & k \leq -1. \end{cases} \tag{2}$$

In fact, MBS has been conceived to be more general and the code supplied by its author allows swapping within larger groups of, e.g., three values $\{3k-2, 3k-1, 3k\}$, $k \geq 1$, etc.

2.2 Embedding invariant

For embedding schemes satisfying the swapping constraint (1) with small subsets $\mathcal{I}_k$, one can construct a useful embedding invariant by replacing all pixel values y_{ij} in $\mathbf{y}$ that lie in $\mathcal{I}_k$ with a fixed value $r_k \in \mathcal{I}_k$. Formally, the mapping $\mathcal{R}$ is thus captured with a parameter $\mathbf{r} \in \mathcal{I}^M$ and acts on each pixel y_{ij} separately

$$\mathcal{R}(y_{ij}) = r_k \text{ whenever } y_{ij} \in \mathcal{I}_k, \tag{3}$$

where we reused the symbol for the invariant mapping for an elementwise mapping on pixel values.

The most useful reference signal is the one with a small $\|\mathcal{R}(\mathbf{y}) - x\|$, where $\|\cdot\|$ is a suitable norm, such as the Frobenius norm. In the most ideal but rarely achievable case, $\mathcal{R}(\mathbf{y}) = \mathbf{x}$, for all $\mathbf{x}, \mathbf{y}$, perfect steganalysis would become possible. In general, if some a priori information about the distribution of cover values in each set $\mathcal{I}_k$ is available, it could (and should) be used to obtain reference signals with a smaller $\|\mathcal{R}(\mathbf{y}) - \mathbf{x}\|$. For example, due to the characteristic shape of the histogram of quantized DCT coefficients in a JPEG file, there will be more DCT coefficients equal to $2k$ than $2k+1$ for $k \geq 0$ and vice versa when $k < 0$. Thus, for algorithms that use LSBR in JPEG domain, examples of which are Jsteg (or in general any algorithm that flips LSBs of DCT coefficients) and OutGuess, we opt for the following reference values $\mathbf{r}$:

$$r_k = \begin{cases} 2k & k \geq 1 \\ 2k+1 & k < 0. \end{cases} \tag{4}$$

Because neither Jsteg (generic LSB flipper in JPEG domain) nor OutGuess modify cover values from $\mathcal{I}_0 = \{0, 1\}$, the mapping $\mathcal{R}$ needs to preserve such values, too :

$$\mathcal{R}(y_{ij}) = y_{ij} \text{ when } y_{ij} \in \mathcal{I}_0. \tag{5}$$

For MBS with swapping sets (2), we use

$$r_k = \begin{cases} 2k-1 & k \geq 1 \\ 2k+1 & k \leq -1 \end{cases} \tag{6}$$

and

$$\mathcal{R}(y_{ij}) = y_{ij} \text{ when } y_{ij} = 0. \tag{7}$$

Note that modern content-adaptive embedding schemes, such as UNIWARD (in both domains), HILL, WOW, MiPOD, or UED use

embedding operations that do not allow construction of the same embedding invariant (reference channel).

3 DETECTION WITH REFERENCE SIGNALS

The use of reference signals in feature based steganalysis was previously investigated in [11]. We wish to stress that this work was limited to the spatial domain only and only one type of embedding operation – LSB replacement. There, the authors proposed to concatenate rich feature vectors extracted from the image and its reference. In particular, the operation of zeroing-out the LSBs :

$$\mathcal{R}(y_{ij}) = 2 \times \lfloor y_{ij}/2 \rfloor \tag{8}$$

was identified as the best way to prepare the reference signal for steganalysis of LSB replacement in the spatial domain. The authors also investigated an alternative approach for forming the feature vector by making the noise residual "aware of pixels' parity" before forming the co-occurrences in the rich model. This was achieved by multiplying the residual extracted at pixel i, j by $1 - \text{LSB}(x_{ij})$. This approach avoided doubling the dimensionality of the parity-aware feature vector. Note that it is not clear, however, how the parity-aware residuals should be implemented within a CNN type of detector because the formation of the residual in a network is not a simple convolution but a cascade of stacked convolutions on non-linearly transformed "residuals." In the rest of this section, we reason about other natural possibilities to incorporate a reference signal within a detector built as a CNN.

One possibility would be to first train a CNN for LSBR and then isolate its front part as a "feature extractor" – the part of the architecture before the classifier, which is typically an Inner-Product (IP) layer or a Multi-Layered Perceptron (MLP). Viewing the network as a feature extractor, the methods described in [11] can be readily applied. This approach, however, merges the reference channel "too late" – in the classifier part of the net – and is thus unlikely to perform well also because the feature extractor cannot distinguish between LSBR and LSB matching (LSBM).

The network could and should learn the convolutional kernels (filters) for the right task from the beginning and in the least constrained manner, following the golden rule of deep learning to let as much as possible to be learned in an end-to-end fashion. An obvious idea to incorporate the reference signal in a CNN is to supply the reference as a second channel. This way, the network can use the reference throughout the entire architecture in a way that is completely driven by the learning algorithm. Many previously proposed deep architectures for steganalysis [3, 22, 32–34, 36], however, initialize the filters in the first layer heuristically, such as setting them to the kernels used in the Spatial Rich Model (SRM) [10], the design choice of the YeNet [34], or DCT bases in J-XuNet [32]. While there is some heuristics reasoning behind these choices for grayscale images, it is not clear how to initialize the kernels for an architecture that accepts an image and its reference on its input. This is one of the reasons why we selected for our study the recently proposed SRNet [2] as its kernels are all initialized randomly as this gives the most freedom to the learning algorithm to determine the most effective combination of kernels. Moreover, based on the results reported in the original paper, it provides state-of-the-art performance in both spatial and JPEG domain.

Here, we only provide a brief summary of the main design choices of the SRNet while referring the reader to the original publication for more details. The SRNet is a residual network with 12 layers employing 3×3 kernels, ReLU activations, and batch normalization. The first eight layers accept unpooled feature maps on their input. This segment of the network learns noise residuals effective for separating the classes of cover and stego images. The first convolutional layer has 64 3×3 kernels, which we modify to 64 $2 \times 3 \times 3$ kernels for a two-channel input as the only modification to incorporate the reference signal. We further note that the selection-channel aware version of the SRNet [2] has not been used because the type of embedding schemes that allow the reference channel introduced in the previous section are not adaptive to cover image content. To be more precise, we are not aware of any competitive, widely studied steganographic algorithms that use LSBR in combination with content adaptivity as this would unnecessarily compromise security.[3]

Network detectors that utilize the reference channel will be abbreviated with an "R-" in front of the detector acronym, such as R-SRNet and R-YeNet.

4 SETUP OF EXPERIMENTS

This section contains the description of the common core of all experiments, the datasets used in our experiments, and the performance evaluation metric.

4.1 Datasets and training

The performance of all detectors was evaluated and contrasted with selected relevant prior art on a dataset created from the union of BOSSbase 1.01 [1] and BOWS2, each containing 10,000 grayscale images resized from their original size 512×512 to 256×256 using 'imresize' with default setting in Matlab. For JPEG experiments, this source was additionally compressed with quality factors 75 and 95.

Randomly chosen 4,000 images from BOSSbase and the entire BOWS2 dataset were used for training with 1,000 BOSSbase images set aside for validation. The remaining 5,000 BOSSbase images were used for testing. In summary, $2 \times 14,000$ cover and stego images were used for training, $2 \times 1,000$ for validation, and $2 \times 5,000$ for testing. This applies to both the spatial and JPEG domain and all detectors. For classifiers trained on rich models, the validation set was added to the training set. Network detectors and the GFR feature set [25] were fed with decompressed JPEG images without rounding to integers.

4.2 Evaluation metric

The detection performance was measured with the total empirical detection accuracy defined as $P_{\text{D}} = 1 - P_{\text{E}}$, where $P_{\text{E}} = \min_{P_{\text{FA}}} \frac{1}{2}(P_{\text{FA}} + P_{\text{MD}})$ is the total classification error probability on the testing set under equal priors with P_{FA} and P_{MD} standing for the false-alarm and missed-detection probabilities. It is worth noting that the default cross entropy loss function used in training neural network detectors maximizes the empirical accuracy since all cover images presented to the network are paired up with the corresponding stego image.

[3] NUGO [24] (Not so Undetectable steGO) was introduced solely for the purpose of studying the effects of content adaptivity known to the detector.

The results reported in Section 5 and 6 are for one random 50/50 split of BOSSbase because it would not be computationally feasible to train all networks on multiple different splits to obtain more statistically robust results. Based on the results reported in [2], the statistical spread of the detection error (scaled to [0,1]) in terms of the mean absolute deviation is 0.002–0.003, which is comparable to what has typically been reported for detectors implemented with rich models and low-complexity classifiers.

5 LSBR IN SPATIAL DOMAIN

This section contains the details of all experiments in the spatial domain, comparison to prior art, and analysis of the results. In particular, we take a closer look at how the R-SRNet adapts the $2 \times 3 \times 3$ filters in the first layer to make use of the reference signal and contrast this with R-YeNet.

As in [11], the stego images for LSBR in spatial domain were prepared by fixing the relative number of flipped bits in the cover – the so-called change rate β. This way, it will be easier for practitioners to relate the reported detectability to a specific payload based on the syndrome coding employed in the specific implementation of this embedding paradigm. In particular, optimal codes[4] would incur an expected change rate β when embedding relative payload of $R = h_2(\beta)$ bits per pixel (bpp), where $h_2(x) = -x \log_2 x - (1-x)\log_2(1-x)$ is the binary entropy function. On the other hand, naive LSBR with no coding would embed only $R = 2\beta$ bpp. In this paper, we focus on small change rates $\beta \in \{0.003, 0.005, 0.01, 0.02, 0.03\}$, which correspond to relative payloads 0.03, 0.045, 0.081, 0.14, and 0.19 bpp with optimal coding.

5.1 Detector training

Both the two-channel R-SRNet and SRNet were trained from randomly initialized weights only for the largest change rate $\beta = 0.03$ for 200k iterations with learning rate (LR) 0.001, which was followed by 100k iterations with LR 0.0001. All smaller payloads were obtained by cascade curriculum training from the largest to the smallest by seeding the network with the detector trained for the next larger change rate. Both the single-channel and the two-channel SRNets were curriculum trained for 100k with LR 0.001 followed by 50k with LR 0.0001. The optimizer Adamax was used with mini-batches of 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[5] and 2×10^{-4} L2 regularization. Filter biases were set to 0.2 and no regularization. The weights in the fully connected classifier layer were initialized with a zero mean Gaussian with standard deviation 0.01 and no bias.

SRNet was compared with YeNet [34] as one of the leading steganalysis detectors in the spatial domain. For the two-channel R-YeNet, the filters in the first layer were initialized with SRM filters in both channels. The network was trained in the same fashion as SRNet, starting with the largest change rate of 0.03 and then cascade-curriculum trained for all other change rates. Initially, we trained for 800k iterations with LR 0.4 and then for an additional 300k iterations with LR 0.08. Curriculum training was carried out for 300k iterations with LR 0.4 and 100k iterations with LR 0.08. The two-channel YeNet was initially trained for the largest payload with 800k iterations with LR 0.4 and 300k iterations with LR 0.08. Curriculum training was carried out for 300k iterations with LR 0.4 and 100k iterations with LR 0.08.

The reference channel for both R-SRNet and R-YeNet was obtained by zeroing-out the LSBs (8). For comparison with prior art, we used exactly the same setup as in [11] – the ensemble classifier with parity-aware 50,856-dimensional rich model (PA-RM). We also implemented the Weighted Stego-image method with bias correction (WSb) [15] as the most accurate detector of this type based on the results reported in Table 5 in [11], where the authors extensively tested the performance of other detectors, such as [5, 7] on both uncompressed images and decompressed JPEGs.

5.2 Performance

The detection accuracy of all detectors appears in Table 1 with Figure 1 contrasting the R-SRNet with previous art. The improvement of R-SRNet w.r.t. prior art (PA-RM as well as R-YeNet) is observed especially for the two smallest change rates (6–9%). The addition of the reference channel helped increase the detection accuracy of R-SRNet P_D by up to 6% with $P_D > 0.98$ for the largest tested change rate 0.03. While the reference channel boosted the accuracy of SRNet by 5–7%, YeNet benefited to a much lesser degree (0.5–2.5%). We hypothesize that this is because both channels were initialized with the same SRM filters, which likely prevented it from learning a more efficient way to incorporate the reference signal. It is worth noting that flipping mere 197 LSBs in 256×256 grayscale images is still detectable with 84% accuracy. Also note that since single-channel networks are blind to pixel parity and thus do not see the difference between LSBR and LSB matching, we can interpret their detection accuracy as that of a non-adaptive LSB matching.

The disadvantage of detectors built with machine learning over WSb is that they need a training set. A mismatch between the training and testing sets may negatively affect their accuracy. On the other hand, if the source of cover images is known, such detectors provide a clear performance advantage over structural detectors.

Figure 2 shows the progress of the training and validation accuracy for LSBR at change rate 0.03 for both tested networks and their versions with the reference channel. The SRNet generally trains faster than YeNet in terms of the number of iterations. Another aspect in which both networks differ is that R-SRNet trains faster than SRNet but R-YeNet trains slower than YeNet.

5.3 Filter analysis

In this section, we analyze the first-layer filters in both channels learned by R-SRNet and R-YeNet and conduct additional experiments to obtain more insight into how the networks use the reference channel.

Most filters learned in R-SRNet trained for the largest change rate of 0.03 are approximately of high-pass nature. The average values

[4] Codes operating on the rate–distortion bound when measuring the distortion as the Hamming distance.
[5] https://arxiv.org/pdf/1502.01852v1.pdf

Table 1: Detection accuracy P_D for LSBR at different change rates β with SRNet, YeNet, and its two-channel versions "R-", parity-aware rich models, and weighted stego-image with bias correction (WSb).

β	Flips	R-SRNet	SRNet	R-YeNet	YeNet	PA-rich	WSb
0.03	1966	**98.32**	93.12	94.97	92.90	97.60	95.74
0.02	1311	**97.35**	91.46	93.46	91.09	95.89	91.22
0.01	655	**94.04**	87.22	88.64	87.12	90.36	79.21
0.005	328	**88.30**	83.40	82.31	81.58	82.08	67.13
0.003	197	**84.16**	78.31	77.59	77.08	75.26	60.69

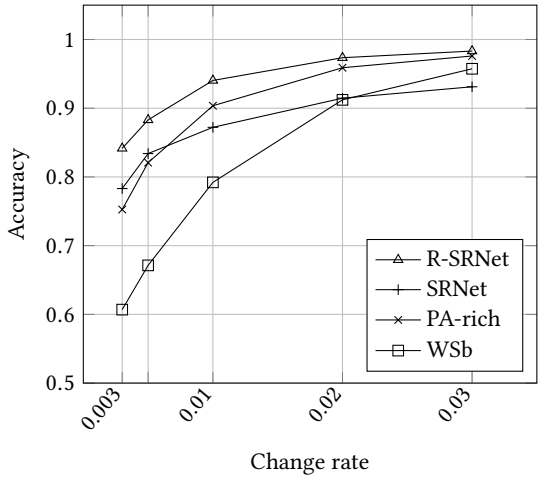

Figure 1: Detection accuracy P_D of LSBR in the spatial domain on the testing set as a function of the change rate β for R-SRNet, parity-aware rich model, and weighted stego-image with bias correction.

of L1-normalized filters were all within $[-0.1, 0.1]$ with the majority symmetrically clustered around zero with a standard deviation of 0.06. This is consistent with interpreting the filters as "noise residual extractors."

To see how the network makes use of the second channel, we inspected the relationship between the two 3×3 filters in each $2\times3\times3$ filter from the first layer. The learned filters in R-SRNet were mostly anti-correlated with a mean (median) normalized correlation -0.2 (-0.173). The two most anti-correlated filters are shown in Figure 3 with correlations -0.9986 and -0.9677, while the two most positively correlated filters were at 0.53 and 0.85.

In contrast to R-SRNet, the filters learned in R-YeNet stayed largely unchanged w.r.t. their initial values: a total of 20 out of 30 3×3 filters exhibited correlations with their initial values larger than 0.95 with only four filters with correlation less than 0.5. Likewise, we observed strong positive correlations between the filters from both channels: 20 with correlation larger than 0.95 and only two with correlation less than 0.5. We hypothesize that the SRM initialization in YeNet is near a local minimum that is difficult for the optimizer to get out of, a problem that may be further exacerbated by the Thresholded Linear Unit (TLU) that nullifies the gradients outside of its threshold. We acknowledge that this a mere hypothesis that we did not further investigate as we consider studying other network architectures for the best use of the reference channel a topic that is outside of the scope of this paper.

Anti-correlated filters (between the two channels) correspond to subtracting the noise residuals extracted from the input image and the reference image. This is intuitively logical and can be interpreted as an operation that further suppresses image content and increases the SNR between the signal of interest, the steganographic embedding changes, and the noise – what is left of the image content. However, perfectly anti-correlated filters (with correlation -1) essentially correspond to convolving just the LSBs of the input image due to the linearity of convolution. It would not be desirable to suppress the content to this degree for all filters because it would limit the detector to information contained only in the LSB plane of the input image.

To obtain further insight, we executed additional experiments in which we forced the filters in the first layer of R-SRNet to be fully anticorrelated with each other and we also initialized the filters in the second channel of R-YeNet as negatives of the SRM filters from the first channel (perfectly anti-correlated). Forcing of the same kernels in both channels of R-SRNet was executed simply by feeding the network with the LSBs of the input image, i.e., in this case, R-SRNet was essentially single-channel. Also note that even when initialized with perfectly anti-correlated filters, during training YeNet will generally learn different filters in both channels because they are independently updated in each iteration – their gradients are generally different.

To better see the impact on the detection accuracy, we ran experiments with a smaller change rate of 0.02. R-YeNet initialized with anti-correlated filters in both channels trained faster and indeed performed slightly better (increase of P_D from 93.46 to 93.95, c.f. Table 1) than when initializing both channels with the same SRM filters. On the other hand, forcing perfectly anti-correlated filters in R-SRNet lead to a markedly worse accuracy of 75.14. As already pointed out above, since perfectly anti-correlated filters imply that the network makes its decision solely based on the LSB plane, the network can detect the embedding only when the LSB plane exhibits learnable structures.

As our final note, we would like to point out that the reference channel allows R-SRNet to better reject content when forming the "noise residuals" in the first seven unpooled layers. In Figure 4, we show the variance of all 16 256×256 feature maps outputted by the last unpooled layer in R-SRNet (black) and in SRNet (white) for the image shown on the right. A larger variance indicates that the feature map contains more "content leftover," which lowers the SNR between the signal of interest – the steganographic embedding

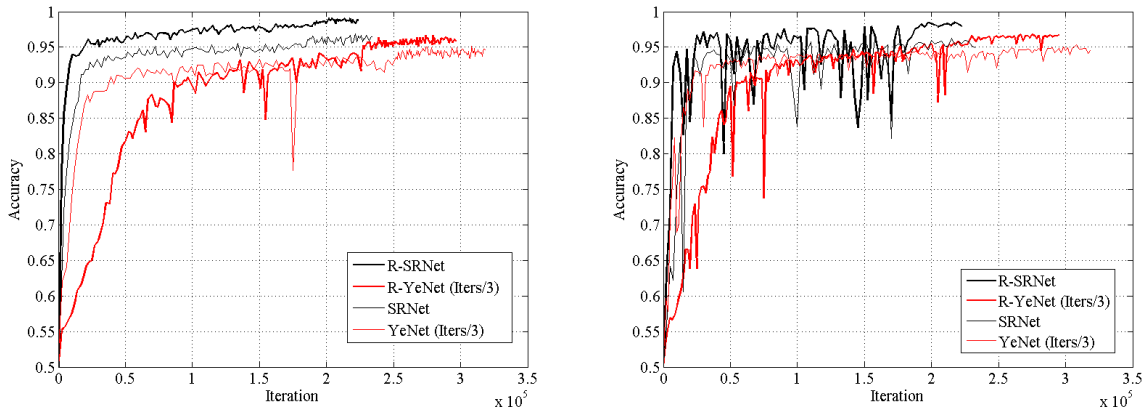

Figure 2: Training (left) and validation (right) accuracy of SRNet, YeNet, and their versions with the reference channel for LSBR in spatial domain at change rate 0.03.

Figure 3: Two examples of 3×3 filters from the first layer of SRNet learned for LSBR at change rate 0.03 that exhibited the largest anticorrelation between both channels. The filters approximately correspond to a diagonal residual and a horizontal second-order residual. Darker / brighter colors correspond to negative / positive values.

changes – and the content of the cover image. While suppression of the content seems necessary for better detection, content suppression in the form of retaining only the LSBs significantly lowers the detection accuracy.

6 JPEG DOMAIN

In this section, we focus on steganographic schemes for JPEG images that allow the type of embedding invariant described in Section 2: a generic LSB flipper (LSBF), OutGuess [21], and model-based steganography [23]. We intentionally opted for the term "generic LSB flipper" (LSBF) for an algorithm whose embedding impact amounts to flipping LSBs of pseudo-randomly selected quantized DCT coefficients while skipping 0's and 1's. This was done to avoid confusion with a specific implementation of LSBR in JPEG domain called Jsteg [28], which uses the same embedding operation but hides the message bits in pixels sequentially selected from the JPEG file, which opens up numerous other possibilities for steganalysis, including the histogram attack [29] or simply analyzing the header. For the same reasons as explained in the beginning of Section 5, the stego images for LSBF were thus created by fixing the change rate β and flipped the LSBs of pseudo-randomly selected βN_{01} DCT coefficients not equal to 0 or 1, where N_{01} is the total number of coefficients not equal to 0 or 1.

Instead of using the OutGuess implementation provided by its creator, Neils Provos, we used its simulator that first embeds the secret payload using (uncoded) LSBR in coefficients not equal to 0 or 1 and then changes some of the unused coefficients, also selected in a pseudo-random fashion, to adjust the histogram of coefficients to its original (cover) form.

The stego images for MBS were obtained with the code supplied by its author, Phil Sallee. For OutGuess and MBS, we fixed the relative payload R in bits per non-zero AC DCT coefficient (bpnzac) because the relationship between R and β for either algorithm depends on the particular cover image. For repeatability of our results, we intend to make the OutGuess and LSBF simulators as well as the embedding algorithm for MBS available for download.

We begin this section with a short study of the effect of the type of the reference channel for the generic LSB flipper. Then, all experiments for the LSBF, OutGuess, and MBS are listed and interpreted. The section is closed with an analysis of the relationship between filters from the two input channels learned in the first layer of the trained R-SRNet.

6.1 Effect of the reference channel

As explained in Section 2, for LSB replacement in the JPEG domain the reference channel prepared with the mapping (4) should intuitively lead to better results because the reference signal $\mathcal{R}(\mathbf{x})$ is closer to the input image $\mathbf{x}$ than $\mathcal{R}(\mathbf{x})$ obtained by zeroing out the LSBs (8). To verify this hypothesis, we executed an experiment with R-SRNet on our dataset (Section 4.1) for JPEG quality 75 and the OutGuess algorithm on a range of payloads (see the next section for the training details). The black bars in Figure 5 show the gain in detection accuracy P_{D} when using the reference (4) instead of zeroing out the LSBs of quantized DCT coefficients. The white bars in Figure 5 show the gain of R-SRNet with reference (4) w.r.t. the single-channel SRNet. We would like to point out that this gain is in the absolute value of the accuracy and not relative. In other

words, a gain of 2.46% means that the values of P_D increased by 0.0246. Having established the superior performance of the reference signal (4), it was selected for all experiments with LSBF and OutGuess.

6.2 Detector training

In JPEG domain, both the single-channel SRNet and R-SRNet were trained from scratch for the largest payload for 200k iterations with LR 0.001 followed by 100k iterations with LR 0.0001, while the cascade curriculum training was run for 100k iterations with LR 0.001 followed by 50k iterations with LR 0.0001. This training procedure was the same for all three steganographic algorithms. The reference image was prepared using (4) for LSBF and OutGuess and using (5) for MBS.

The performance of R-SRNet was contrasted with classifiers implemented using the low-complexity linear classifier [4] with the JPEG Rich Model (JRM) [17] and the Gabor Filter Residual (GFR) [25] model as well as their versions in which the feature was augmented with a feature extracted from the same reference signal (4), essentially doubling the dimensionality of each rich model (R-JRM and R-GFR). The GFR model was selected as one of the most effective JPEG-phase-aware feature set for detection of modern embedding algorithms[6] while JRM is typically effective against old embedding algorithms, such as the three schemes studied in this paper.

We also included in our tests the results obtained with the Generalized Likelihood Ratio (GLR) test proposed for Jsteg (LSBF) in [26] and for OutGuess in [27]. Similar to the machine-learning detectors, these GLR detectors need to be informed by the suspected payload embedded in the analyzed image. Also, a payload correction needs to be added for OutGuess to adjust for the increased change rate (equivalent payload) due to the histogram correction step.

6.3 Performance

First, we take a look at how R-SRNet benefits from the second channel. Table 2 contrasts the accuracy of R-SRNet with the single-channel SRNet for all three embedding algorithms and two JPEG quality factors. Two more payloads (0.1 and 0.05 bpnzac) were added for MBS to show how the accuracy saturates with increasing message length. For LSBF, the second channel increases the accuracy of R-SRNet by up to 7.4% for small change rates. As expected, for easier detection cases when the accuracy approaches 100%, the gain is smaller. For OutGuess, we observed an even stronger effect of the reference channel, boosting the accuracy by up to 12.4% for small payloads and JPEG quality 95. Finally, for MBS the largest observed gain was almost 6% again, generally, for smaller payloads.

Similar to the spatial domain, the reference signal helps not only the performance but also speeds up the convergence. This was observed for all three JPEG steganographic algorithms and all tested payloads / change rates. Figure 6 contrasts the detection accuracy of R-SRNet with classifiers trained on R-JRM and R-GFR models for JPEG quality 75. While low-complexity linear classifiers trained with JRM features detect all three algorithms better than with GFR features, the latter benefits significantly more from the reference channel (4) and is clearly the better detector for MBS (see Figure 6 right). For OutGuess, R-JRM is better than R-GFR for all payloads except for the largest payload, and for LSBF both features with reference channels exhibit approximately the same performance. The figures for LSBF and OutGuess (left and middle) also show the performance of the GLR [26, 27]. Overall, the two-channel R-SRNet as well as the single-channel SRNet clearly outperform all tested previous art.

[6] A slightly better detection may be obtained with the correctly symmetrized GFR feature [31].

Table 2: Detection accuracy with SRNet with reference channel (R-SRNet) and a single-channel for LSBR in JPEG domain (LSBF) at different change rates β and for OutGuess and MBS for a range of relative payloads R in bpnzac.

	LSBF QF 75		LSBF QF 95	
β	R-SRNet	SRNet	R-SRNet	SRNet
0.03	99.96	99.13	99.87	99.50
0.02	99.65	98.42	99.64	98.54
0.01	96.93	92.98	97.95	94.07
0.005	89.10	83.16	91.90	85.74
0.003	80.84	74.42	84.75	77.37
	OutGuess QF 75		OutGuess QF 95	
R	R-SRNet	SRNet	R-SRNet	SRNet
0.03	99.11	96.08	99.50	97.21
0.02	97.18	90.36	98.71	94.79
0.01	90.14	81.44	94.75	84.94
0.005	79.39	71.00	84.83	74.20
0.003	72.86	64.50	79.37	66.95
	MBS QF 75		MBS QF 95	
R	R-SRNet	SRNet	R-SRNet	SRNet
0.1	99.76	98.81	99.86	99.52
0.05	96.90	93.84	98.99	96.70
0.03	91.09	85.76	95.85	90.93
0.02	84.79	80.06	91.44	85.74
0.01	73.48	68.88	81.25	75.59
0.005	64.29	60.65	69.37	65.77
0.003	59.08	57.37	62.61	59.99

6.4 Filter analysis

We now take a closer look at the filters learned in the first convolutional layer of R-SRNet and their relationship measured by normalized correlation between the two 3×3 filters from each 2×3×3 filter. While in the spatial domain, the filters were largely anti-correlated, in JPEG domain the corresponding filters in R-SRNet exhibit more positive correlations. This is documented in Table 3 that shows the minimum, average, and maximum correlations across all 64 filters. The correlations tend to increase with decreasing payload R and are more positive for OutGuess than for LSBF, and exhibit even more positive correlations for MBS.

As in the spatial domain, most filters in either channel of the R-SRNet trained for all three embedding algorithms were observed to be of high-pass nature. Compared to the filters learned for LSBR

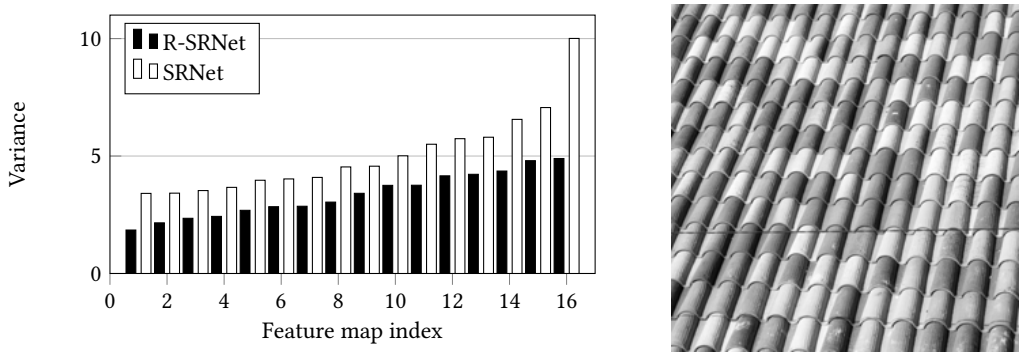

Figure 4: Variance of feature maps outputted by the last unpooled layer in SRNet (black) and R-SRNet on the input image shown on the right for LSBR in the spatial domain for change rate 0.02.

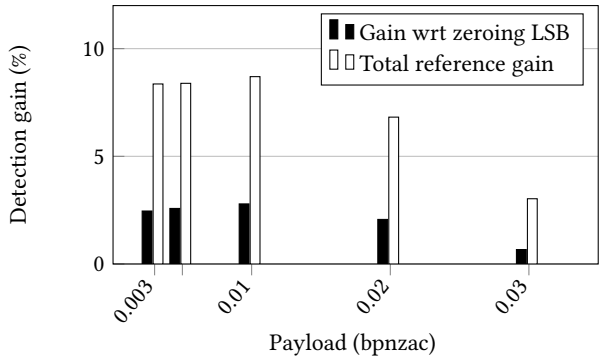

Figure 5: Black: Gain in detection accuracy (%) on the testing set with R-SRNet when using the reference (4) instead of zeroing out the LSBs of quantized DCT coefficients as a function of relative payload in bpnzac for OutGuess at JPEG quality 75. White: Gain of R-SRNet with reference (4) w.r.t. single-channel SRNet for the same setup.

in the spatial domain, the means of L1-normalized filters are more tightly clustered around zero with standard deviation of 0.03–0.04.

6.5 Stego source mismatch

The three embedding algorithms in JPEG domain studied above share certain similarities. OutGuess and LSBF use the same embedding operation but OutGuess makes additional changes of the same nature to the image to restore its histogram to the original form. In contrast, while the embedding operation of MBS is identical to that of OutGuess or LSBF for negative DCT coefficients, it is different for positive coefficients. Because of these similarities and differences, in this section we test how well a detector trained on one embedding algorithm can detect stego images outputted by the other two algorithms. In other words, we study a specific form of a stego-source mismatch.

Table 4 shows the detection accuracy of R-SRNet trained on one algorithm and tested on another. The payloads / change rates for each scheme were selected to force the same level of detection accuracy for each individual algorithm (around 90%). As expected, the mismatch in the stego source between LSBF and OutGuess is negligible. The detector of MBS, however, exhibits a large missed detection rate on LSBF and OutGuess, which is again to be expected because the reference channel is not prepared correctly and also because the embedding scheme is different.

Table 3: Minimum, average, and maximum correlation between 3×3 filters from the first layer from both channels in R-SRNet trained for two payloads R and three embedding algorithms. JPEG quality 75.

R	0.003			0.001		
	min	avg	max	min	avg	max
LSBF	−0.74	−0.12	0.83	−0.90	0.03	0.98
OG	−0.88	0.05	0.93	−0.90	0.14	0.99
MBS	−0.83	0.29	0.97	−0.86	0.39	1.00

Table 4: Detection accuracy of R-SRNet when training on one embedding algorithm and testing on another (JPEG quality 75). The generic LSB flipper (LSBF) was trained for change rate $\beta = 0.005$, OutGuess for payload $R = 0.01$ bpnzac, and MBS for payload $R = 0.03$ bpnzac.

	Tested on		
Trained	LSBF	OG	MBS
LSBF	89.10	89.64	56.34
OG	89.37	90.14	58.17
MBS	68.33	68.75	91.09

7 CONCLUSIONS

This paper revisits an old topic – detection of the early steganographic schemes with embedding operations that swap values from small disjoint subsets of the cover dynamic range, an example of which is the LSB replacement. Steganography that uses such embedding operations allows construction of reference signals that

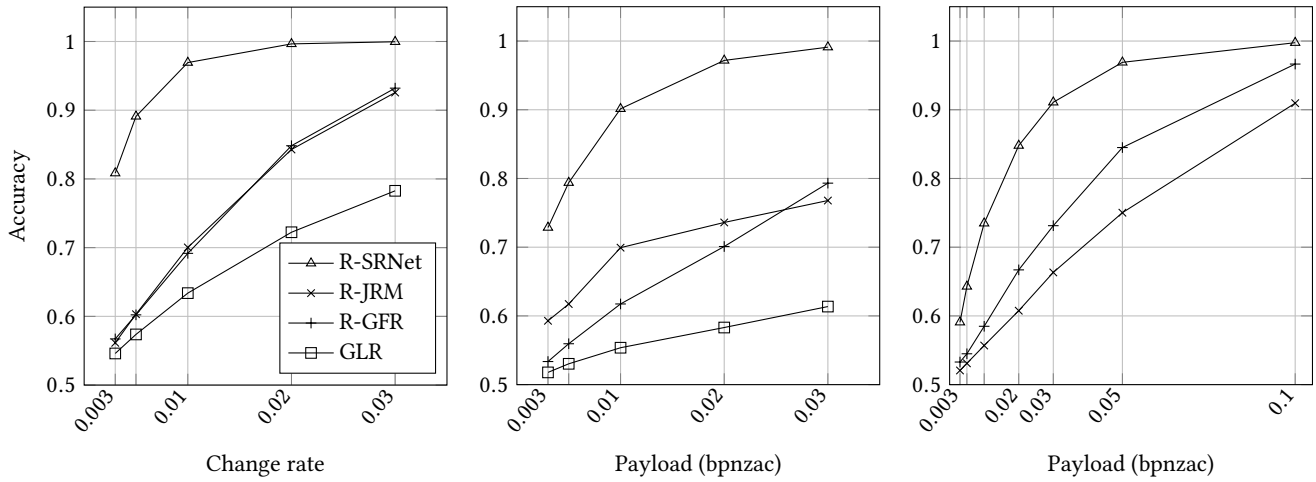

Figure 6: Detection accuracy as a function of change rate / payload for R-SRNet, R-JRM, and R-GFR for LSBF (left), OutGuess (middle), and MBS (right), for JPEG quality 75. For LSBF and OutGuess, we also include the detection performance of the GLR [26, 27]. The legend is common across all three charts.

are close to the input image but are not sensitive to embedding modifications. As shown in the past, augmenting a rich feature with a feature computed from the reference signal (parity-aware rich models) can significantly improve detection of LSB replacement with feature-based classifiers.

This paper extends the prior art in two ways. First, we consider more general type of embedding operations than replacement of LSBs in the spatial domain by showing how the reference signal should be prepared for a generic LSB flipper in the JPEG domain (LSBF), example of which is Jsteg. Second, we study how this reference should be used within the novel detection paradigm – deep convolutional neural networks. Since the network learns the image representation as well as the steganalysis classification jointly, we present the reference image as a second input channel to the network. We work with the recently proposed SRNet because all its filters are randomly initialized, which allows the network to discover a way to incorporate the second channel via data driven end-to-end training. Further research seems to be needed on how to incorporate reference channels in network architectures with heuristically initialized or fixed filters in the first layer, such as the YeNet.

The merit of the proposed ideas is shown on experiments with LSBR in spatial domain, OutGuess, MBS, and LSBF. The detection accuracy of the two-channel SRNet (R-SRNet) is compared with classifiers trained on rich models as well as rich models augmented with features computed from the same reference signal presented to the network. Additionally, we report the results for other types of detectors, including structural attacks, the weighted stego-image detector, and detectors for LSBF and OutGuess constructed with statistical signal detection tools from models of DCT coefficients.

Steganalysis with reference images is a special case of a more general, modern topic of detection with side-information at the detector. While most recent work focused on side-information in the form of the selection channel (the embedding change probabilities) to improve detection of content-adaptive steganography, there are many other types of potentially useful reference signals that could be formed from the input image, such as alternative color space representations or reference signals obtained by filtering the input image. Investigating these directions will be a part of our future effort.

Finally, we stress that our study focused only on embedding schemes with a swapping constraint (see Section 2.1) that permit the type of reference signals investigated in this paper. Modern content-adaptive embedding schemes, such as UNIWARD (in both domains), HILL, WOW, MiPOD, or UED use embedding operations that do not allow construction of the same embedding invariant (reference channel).

ACKNOWLEDGMENTS

The work on this paper was supported by NSF grant No. 1561446. The authors would like to thank Remi Cogranne for sharing the code for the GLR test for Jsteg and OutGuess.

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] M. Boroumand, M. Chen, and J. Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5):1181–1193, May 2019.

[3] 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.

[4] 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.

[5] R. Cogranne, C. Zitzmann, L. Fillatre, F. Retraint, I. Nikiforov, and P. Cornu. A cover image model for reliable steganalysis. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, Lecture Notes in Computer Science, pages 178–192, Prague, Czech Republic, May 18–20, 2011.

[6] S. Dumitrescu and X. Wu. LSB steganalysis based on higher-order statistics. In A. M. Eskicioglu, J. Fridrich, and J. Dittmann, editors, *Proceedings of the 7th ACM*

Multimedia & Security Workshop, pages 25–32, New York, NY, August 1–2, 2005.

[7] L. Fillatre. Adaptive steganalysis of least significant bit replacement in grayscale images. *IEEE Transactions on Signal Processing*, 60(2):556–569, 2011.

[8] J. Fridrich and M. Goljan. On estimation of secret message length in LSB steganography in spatial domain. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VI*, volume 5306, pages 23–34, San Jose, CA, January 19–22, 2004.

[9] J. Fridrich, M. Goljan, and R. Du. Detecting LSB steganography in color and gray-scale images. *IEEE Multimedia, Special Issue on Security*, 8(4):22–28, October–December 2001.

[10] 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.

[11] J. Fridrich and J. Kodovský. Steganalysis of LSB replacement using parity-aware features. In M. Kirchner and D. Ghosal, editors, *Information Hiding, 14th International Conference*, volume 7692 of Lecture Notes in Computer Science, pages 31–45, Berkeley, California, May 15–18, 2012.

[12] A. D. Ker. A general framework for structural analysis of LSB replacement. In M. Barni, J. Herrera, S. Katzenbeisser, and F. Pérez-González, editors, *Information Hiding, 7th International Workshop*, volume 3727 of Lecture Notes in Computer Science, pages 296–311, Barcelona, Spain, June 6–8, 2005. Springer-Verlag, Berlin.

[13] A. D. Ker. Resampling and the detection of LSB matching in color bitmaps. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 1–15, San Jose, CA, January 16–20, 2005.

[14] A. D. Ker. Fourth-order structural steganalysis and analysis of cover assumptions. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, pages 25–38, San Jose, CA, January 16–19, 2006.

[15] A. D. Ker and R. Böhme. Revisiting weighted stego-image steganalysis. In E. J. Delp, P. W. Wong, J. Dittmann, and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819, pages 5 1–17, San Jose, CA, January 27–31, 2008.

[16] J. Kodovský and J. Fridrich. Calibration revisited. In J. Dittmann, S. Craver, and J. Fridrich, editors, *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 63–74, Princeton, NJ, September 7–8, 2009.

[17] J. Kodovský and J. Fridrich. Steganalysis of JPEG images using rich models. In A. Alattar, N. D. Memon, and E. J. Delp, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2012*, volume 8303, pages 0A 1–13, San Francisco, CA, January 23–26, 2012.

[18] B. Li, W. Wei, A. Ferreira, and S. Tan. ReST-Net: Diverse activation modules and parallel subnets-based CNN for spatial image steganalysis. *IEEE Signal Processing Letters*, 25(5):650–654, May 2018.

[19] X. Li, T. Zeng, and B. Yang. Detecting LSB matching by applying calibration technique for difference image. In A. D. Ker, J. Dittmann, and J. Fridrich, editors, *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 133–138, Oxford, UK, September 22–23, 2008.

[20] Q. Liu. Steganalysis of DCT-embedding based adaptive steganography and YASS. In J. Dittmann, S. Craver, and C. Heitzenrater, editors, *Proceedings of the 13th ACM Multimedia & Security Workshop*, pages 77–86, Niagara Falls, NY, September 29–30, 2011.

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

[22] 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.

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

[24] P. Schöttle, S. Korff, and R. Böhme. Weighted stego-image steganalysis for naive content-adaptive embedding. In *Fourth IEEE International Workshop on Information Forensics and Security*, Tenerife, Spain, December 2–5, 2012.

[25] X. Song, F. Liu, C. Yang, X. Luo, and Y. Zhang. Steganalysis of adaptive JPEG steganography using 2D Gabor filters. In P. Comesana, J. Fridrich, and A. Alattar, editors, *3rd ACM IH&MMSec. Workshop*, Portland, Oregon, June 17–19, 2015.

[26] T. Thai, R. Cogranne, and F. Retraint. Statistical model of quantized DCT coefficients: Application in the steganalysis of Jsteg algorithm. *Image Processing, IEEE Transactions on*, 23(5):1–14, May 2014.

[27] T. H. Thai, R. Cogranne, and F. Retraint. Optimal detection of OutGuess using an accurate model of DCT coefficients. In *Sixth IEEE International Workshop on Information Forensics and Security*, Atlanta, GA, December 3–5, 2014.

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

[29] A. Westfeld and A. Pfitzmann. Attacks on steganographic systems. In A. Pfitzmann, editor, *Information Hiding, 3rd International Workshop*, volume 1768 of Lecture Notes in Computer Science, pages 61–75, Dresden, Germany, September 29–October 1, 1999. Springer-Verlag, New York.

[30] J. M. Whitaker and A. D. Ker. Steganalysis of overlapping images. In A. Alattar and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2015*, volume 9409, pages 2601–2615, San Francisco, CA, February 8–12, 2015.

[31] C. Xia, Q. Guan, X. Zhao, Z. Xu, and Y. Ma. Improving GFR steganalysis features by using Gabor symmetry and weighted histograms. 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.

[32] 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.

[33] 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.

[34] 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.

[35] M. Yedroudj, F. Comby, and M. Chaumont. Yedroudj-net: An efficient CNN for spatial steganalysis. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2092–2096, April 2018.

[36] 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, May 2018.

[37] C. Zitzmann, R. Cogranne, F. Retraint, I. Nikiforov, L. Fillatre, and P. Cornu. Statistical decision methods in hidden information detection. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, Lecture Notes in Computer Science, pages 163–177, Prague, Czech Republic, May 18–20, 2011.